SRI International

(2)

COMPUTER VISION RESEARCH AND ITS APPLICATIONS TO AUTOMATED CARTOGRAPHY

Fourth and Fifth Semiannual Technical Reports
Covering the period June 11, 1984 to June 10, 1985

Contract Amount:	$3,654,877
Effective Date:	10 December 1982
Expiration Date:	30 September 1985

September 1985

By: Martin A. Fischler, Program Director
Project Leader, (415) 859-5106

Artificial Intelligence Center
Computer Science and Technology Division

Prepared for:

Defense Advanced Research Projects Agency
1400 Wilson Boulevard
Arlington, Virginia 22209

Attention: Lt.Col. Robert L. Simpson, Jr., Program Manager
Information Processing Techniques Office

SRI Project 5355
Contract No. MDA903-83-C-0027
ARPA Order No. 3862 and AMD 8
Program Code No. 3D30, Program Element 61101E

This document has been approved for public release and sale; its distribution is unlimited.

SRI International

333 Ravenswood Ave. • Menlo Park, CA 94025
(415) 326-6200 • TWX: 910-373-2046 • Telex: 334-486

bb

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER 4th & 5th Semiannual Technical	2. GOVT ACCESSION NO. AD-A159 435	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) COMPUTER VISION RESEARCH AND ITS APPLICATIONS TO AUTOMATED CARTOGRAPHY		5. TYPE OF REPORT & PERIOD COVERED Combined Semiannual Technical 6/11-12/10/84, 12/11/84-6/10/85 6. PERFORMING ORG. REPORT NUMBER 5355 4th & 5th Semiann. Tech.
7. AUTHOR(s) Martin A. Fischler		8. CONTRACT OR GRANT NUMBER(s) MDA903-83-C-0027
9. PERFORMING ORGANIZATION NAME AND ADDRESS SRI International 333 Ravenswood Avenue Menlo Park, California 94025		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS Program Code No. 3D30 Program Element 61101E
11. CONTROLLING OFFICE NAME AND ADDRESS Defense Advanced Research Projects Agency 1400 Wilson Boulevard Arlington, Virginia 22209		12. REPORT DATE September 1985 13. NUMBER OF PAGES 196
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office) DCASMA, San Francisco 1250 Bayhill Drive San Bruno, CA 94066		15. SECURITY CLASS (of this report) Unclassified 15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

image understanding, computer vision, automated cartography, feature extraction, stereo compilation

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

-The SRI Image Understanding program is a broad effort spanning the entire range of machine vision research. Its three major concerns are: (1) to develop an understanding of the physics and mathematics of the vision process, (2) to develop a knowledge-based framework for integrating and reasoning about sensed (imaged) data, and (3) to develop a machine-based environment for effective experimentation, demonstration, and evaluation of our theoretical results, as well as providing a vehicle for technology transfer. This report describes recent process in all three

ABSTRACT (continued)

areas. In particular, we describe progress in constructing and testing a state-of-the-art automated system for stereo compilation, new approaches to extracting depth and structural information from imaged data, a knowledge-based system for feature extraction, and tools for scene modeling and interaction with a machine terrain data base.

CONTENTS

CONTENTS (cont'd)

PREFACE

This report combines the semiannual technical reports for the periods June 11, 1984 through December 10, 1984 and December 11, 1984 through June 10, 1985.

THE SRI IMAGE UNDERSTANDING RESEARCH PROGRAM

M.A. Fischler (Principal Investigator)

I INTRODUCTION

The goal of this research program is to obtain solutions to fundamental problems in computer vision; particularly to such problems as stereo compilation, feature extraction, and general scene modeling that are relevant to the development of an automated capability for interpreting aerial imagery and the production of cartographic products.

To achieve this goal, we are engaged in investigations of such basic issues as image matching, partitioning, representation, and physical modeling (shape from shading, texture, and optic flow; material identification; recovery of imaging and illumination parameters such as "vanishing points," "camera parameters," and illumination source location; edge classification; etc.). However, it is obvious that high-level, high-performance vision requires the use of both intelligence and stored knowledge (to provide an integrative framework), as well as an understanding of the physics and mathematics of the imaging process (to provide the basic information needed for a reasoned interpretation of the sensed data). Thus, a significant portion of our work is devoted to developing new approaches to the problem of "knowledge-based vision." Finally, vision research cannot proceed without a means for effective implementation, demonstration, and experimental verification of theoretical concepts; we have developed an environment in which some of the newest and most effective computing instruments can be employed for these purposes.

The research results described in this report are partitioned into three topic areas: (1) three-dimensional scene modeling and stereo reconstruction; (2) feature extraction: scene partitioning and semantic labeling; and (3) interactive scene modeling and knowledge-base construction.

II THREE-DIMENSIONAL SCENE MODELING AND STEREO RECONSTRUCTION

Our goal in this research area is to develop automated methods for producing a 3-D scene model from several images recorded from different viewpoints. The standard approach to this problem is to use stereo compilation -- a technique that involves finding pairs of corresponding scene points in two images (which depict the scene from different spatial locations) and using triangulation to determine scene depth. Various factors associated with viewing conditions and scene content can cause the matching process to fail; these factors include occlusion, projective or imaging distortion, featureless areas, and repeated or periodic scene structures. Some of these problems can only be solved by providing the machine with a global context for dealing with the missing or ambiguous information. Thus, an important component of this research effort, discussed in the section on interactive scene modeling, is to devise machinery by which a human operator can simply and effectively provide the needed information. In the remainder of this section we limit our discussion to direct approaches -- more effective methods for image matching, interpolation for filling in "holes" caused by matching failure, and some exciting and radically new methods for 3-D modeling.

A. Baseline Stereo System

As a framework for integration and evaluation of our research in modeling 3-D scene geometry, as well as a vehicle for technology transfer, we have implemented a complete "state-of-the-art" stereo system. This system, described in Appendix A [6,7], is capable of producing a dense 3-D scene model from stereo pairs of intensity images. Included in the appendix are results of testing the system on a number of significant data sets. We believe that the current version of this fully automatic system is comparable to the best of the semiautomatic (human-assisted) systems now in operational use.

B. New Methods for Stereo Compilation

As we previously indicated, the conventional approach to recovering scene geometry from a stereo pair of images is based on the matching of distinctive scene features, as well as on the satisfaction of constraints imposed by the viewing geometry (e.g., the epipolar constraint). Typically, three steps are required: (1) determination of the relative orientation of the two images, (2) computation of a sparse depth map, and (3) derivation of a dense depth map for the given scene.

In the first step, points corresponding to unmistakable scene features are identified in each of the images. The relative orientation of the two images is then calculated from these points. This is, in part, an unconstrained matching task. Corresponding image features must be found. Without a priori knowledge, such a matching procedure knows neither the approximate location (in the second image) of a feature found in the first image, nor the appearance of that feature. However, it is often the case that appearance will vary little between images and that they were taken from similar positions relative to the scene.

Recovery of the relative orientation of the images reduces the computation of a sparse depth map from unconstrained two-dimensional matching to constrained one-dimensional matching. The quest for a scene feature identified in the first image is reduced to a one-dimensional search along an (epipolar) line in the second image. Identification of this feature in the second image makes it possible to calculate the feature's disparity and, hence, its relative scene depth.

Identification of corresponding points in the two images is typically based on correlation techniques. Area-based correlation processes may be applied directly to the raw image irradiances or to images that have been preprocessed in some manner. Edges (identified by the zero crossings of the Laplacian of their image irradiances) have also been used to obtain correspondences.

The outcome of this second step is a sparse map of the scene's relative depth at those points that were identified in both images of the stereo pair.

A sparse depth map does not define the scene topography. The third and final step in recovering the topography of the scene is "filling in" this sparse map to obtain a dense depth map of the scene. Typically, a surface interpolation or approximation method is used as a means of calculating the dense depth map from its sparse counterpart. A surface approximation model may be formulated to provide desirable image properties (such as the lack of additional zero crossings -- in the Laplacian of the image irradiances -- that are artifacts of the surface approximation model), but, often, the surface model is based on a priori requirements for the fitted surface, such as smoothness.

The problems encountered in the first two steps -- recovery of the relative orientation of the images and computation of the sparse depth map -- are dominated by the problems of image matching. False matches that arise from repetitive scene structures, such as windows of a building, or from image features that are not distinctive (at least on the basis of local evidence) occur more frequently in the unconstrained matching environment than in the constrained environment. In recovering the relative orientation of the images, we can use redundant information in an effort to reduce the influence of false matches; this is more difficult in the case when the sparse depth map is computed. Furthermore, we have little choice as to which features we may use for sparse depth mapping; if we choose not to use a feature, we cannot recover the relative depth at that scene point (without invoking semantic or contextual knowledge).

The selection of suitable features for determining image correspondence is difficult in itself. Correlation techniques embed assumptions that are often violated by the best image features. Area-based correlation techniques usually reflect the premise that image patches are of a scene structure that is positioned at one distinct depth, whereas edges that arise at an object's boundaries are surrounded by surfaces at different scene depths. Edge-based techniques are based on the assumption that an edge found in one image is not "moved" by the change in viewing position of the second image, whereas zero crossings found at boundaries of objects whose surface gradients are tangential to the line of sight contradict this assumption. These would seem minor problems, were it not for the accuracy required of the matching process. Often, the spatial resolution of disparity measurements must be better than the image's spatial resolution. Stereo matching sometimes requires features with properties that are incompatible with what is practical in realistic situations.

The third step, derivation of a dense depth map from a sparse one, is still far short of having an adequate solution. Most approaches employ "blind" interpolation, since no effective methods are currently in use for extracting depth from the irradiance data in the individual images of the stereo pair.

In summary, we see that the most demanding steps in the stereo process are the final two: computation of a sparse depth map, and derivation of its dense counterpart. In Appendix B [13], we describe a new approach to stereo compilation that involves combining these steps to recover a dense relative-depth map of the scene directly from the image data. We use image irradiance profiles as input to an integration routine that returns the corresponding dense relative-depth profile. This procedure neither matches image points (at least not in the conventional sense), nor does it "fill in" data to obtain the dense depth map. It avoids the need to make the restrictive assumptions usually required for stereo image matching, and it directly uses the image irradiance data in recovering the dense depth map.

C. New Methods for 3-D Modeling Using Methods Which Do Not Depend On Stereo Correspondence

We have noted the fact that it will not always be possible to find corresponding scene points in the two images of a conventional stereo pair, and yet, to recover a dense scene model, we need to determine the depth at every scene point. Since interpolation will not always provide an acceptable answer when matching fails, we are investigating a number of new techniques for recovering scene depth that do not require establishing stereo correspondence.

A significant body of work exists in the area of extracting depth from the shading and texture visible in a single image. However, these different techniques make a variety of distinct assumptions about the nature of the scene, the illumination, and the imaging geometry. In Appendix C [14], we show that the distinct assumptions employed by each of these different schemes must be equivalent to providing a second (virtual) image of the original scene, and that all of these different approaches can be translated into a conventional stereo formalism. In particular, we show that it is frequently possible to structure the problem as that of recovering depth from a stereo pair consisting of a conventional perspective image (i.e., the original image) and an orthographic image (the virtual image). We also provide a new algorithm needed to accomplish this type of stereo-reconstruction task.

In Appendix D [10] we show how focal gradients (image "blur"), resulting from the limited depth of field inherent in most optical systems, can be used to recover scene depth. The advantages of this technique are that it is fast, computationally simple, makes no special assumptions about the scene, and avoids the stereo-matching problem. Mathematical analysis and experiments indicate that the accuracy achievable by this technique is comparable to what can be expected from the use of stereo disparity or motion parallax in determining scene depth.

For most purposes concerned with the analysis of imaged data, determination of an array of depths (e.g., as obtained by conventional stereo methods) is only the first step in the construction of a scene description. The conventional approach next compiles largely continuous surfaces from the discrete depth information and then attempts to partition these surfaces into coherent 3-D objects. Aside from some still unsolved theoretical problems, this process is computationally expensive and time consuming. In Appendix E [2], we describe a new method for using camera motion through a scene to obtain a 3-D model in which higher level scene attributes are directly accessible. This technique is based on considering a dense sequence of images as forming a solid block of data. Slices through this solid at appropriately chosen angles intermix time and spatial data in such a way as to simplify the partitioning problem: these slices have more explicit structure than the conventional images from which they were obtained. We believe that this work is a very important development; it offers a completely new and direct method for accessing information about scene objects without requiring a completely bottom-up analysis process.

III FEATURE EXTRACTION: SCENE PARTITIONING AND SEMANTIC LABELING

Creating a scene description from a photograhic image requires the ability to perform two basic operations: (a) partitioning the image into independent or coherent pieces, and (b) assigning names or semantic labels to these pieces.

The partitioning operation, necessary to reduce the computational complexity of the subsequent scene-analysis steps, has proven to be extremely difficult to accomplish: the performance of automated systems is still far inferior to that of humans. In part, this disparity in performance occurs because humans appear to employ contextual knowledge and past experience in such tasks, while most available computational techniques employ only the local intensity patterns visible in the image, i.e., they perform "syntactic partitioning." For practical as well as theoretical reasons, we have been pursuing an investigation (1) to determine the competence limits of a purely syntactic approach to partitioning and, simultaneously, (2) to construct an operational system that approaches these limits. This investigation is nearing completion and has resulted in a very high performance system that will be described in a paper now in preparation [8].

In Appendix F [1], we describe one of a number of on-going investigations that attempt to provide a theoretical basis for the partitioning process. In this paper, Barnard explores the idea that partitioning decisions result in alternative descriptions of a scene, and that the preferred partitioning is the one that provides the "simplest" description. In a paper by Fischler and Bolles [3], partitioning is viewed as an explanation of how the image is related to the scene from which it was derived; it is shown that completeness and stability of explanation, as well as simplicity, are useful partitioning criteria since these attributes are necessary for an explanation to be believable.

In Appendix G [5], we describe an approach to the problem of converting a syntactically partitioned image (e.g., one provided by Laws' segmentation system) into a semantic description. This work has resulted in a system that can extract cultural objects from aerial imagery; it employs geometric reasoning to identify semantically significant arrangements of straight line segments in the borders of the supplied partition. Emphasis is placed on using generic models characterizing significant kinds of geometric relationships and shapes, thereby avoiding the well-known drawbacks inherent in the use of specific object templates. An important feature of this system (still under development) is the generation of an explanation for any detected discrepancy between the hypothesized object models and the initial partition. In principle, this technique should permit intelligent compensation for anomalies due to imaging or environmental effects that would be recognized by a well-briefed human analyst; for example, the system should be able to identify two contrasting regions of a peaked roof as belonging to a single house based on illumination effects consistent with the known sun position. The ability of this system to explain its decisions in terms of deviations of sensed data from stored models appears to offer an effective mechanism for understanding the operation of the system and, simultaneously, a basis for improving its performance.

IV INTERACTIVE SCENE MODELING AND KNOWLEDGE-BASE CONSTRUCTION

Our intent in this effort is to develop a system framework for allowing higher-level knowledge to guide and integrate the detailed interpretation of imaged data by autonomous scene-analysis techniques. Such an approach allows symbolic knowledge, provided by higher-level knowledge sources, to control automatically the selection of appropriate algorithms, adjust their parameters, and apply them in the relevant portions of the image. More significantly, we are attempting to provide an efficient means for supplying and using qualitative knowledge about the semantic and physical structure of a scene so that the machine-produced interpretation, constrained by this knowledge, will be consistent with what is generally true of the overall scene structure, rather than just a good fit to locally applied models.

An important component of our approach is to design a means for a human operator simply and effectively to provide the machine with a qualitative scene description in the form of a semantically labeled 3-D "sketch." This capability for effective communication between a human and a machine about the three-dimensional world requires both appropriate graphics tools and an ability on the part of the machine for both spatial reasoning and some semantic "understanding." The importance of this work derives from the fact that a major difficulty in automating the image-interpretation process is the inability of current computer systems to deduce, from the visible image content, the general context of the scene (e.g., urban or rural; season of the year; what happened immediately before, and what will happen immediately after, the image was viewed by the sensor) -- the knowledge-base and reasoning required for such an ability is well beyond what the state of our art can hope to accomplish over (at least) the next 5 years. Thus, our work is intended to provide a means by which a human can supply, to a task-oriented program, the high-level overview the program needs for its analysis of a given scene, but cannot acquire by itself.

A. The Representation and Modeling of Natural Forms

Our research in this area addresses three related problems: (1) representing natural shapes such as mountains, vegetation, and clouds; (2) computing such descriptions from image data; and (3) interactively providing the machine with a description of natural forms as a way of building an internal knowledge data base. The first step towards solving these problems is to obtain a model of natural surface shapes.

A model of natural surfaces is extremely important because we face problems that seem impossible to address with standard descriptive computer-vision techniques. How, for instance, should we describe the shape of leaves on a tree? Or grass? Or clouds? When we attempt to describe such common, natural shapes using standard representations, the result is an unrealistically complicated model. Furthermore, how can we extract 3-D information from the image of a textured surface when we have no effective models that describe natural surfaces and how they evidence themselves in the image? The lack of such a 3-D model has restricted image texture descriptions to being ad hoc statistical measures of the image intensity surface.

Fractal functions, a novel class of naturally arising functions, are a good choice for modeling natural surfaces because many basic physical processes (e.g., erosion and aggregation) produce a fractal surface shape, and because fractals are widely used as a graphics tool for generating natural-looking shapes. Additionally, in a survey of natural imagery, we found that a fractal model of imaged 3-D surfaces furnishes an accurate description of both textured and shaded image regions, thus providing validation of this physics-derived model for both image texture and shading.

Progress relevant to computing 3-D information from imaged data by use of a fractal model is described in Pentland [9]. A test has been derived to determine whether or not the fractal model is valid for a particular set of image data, an empirical method for computing surface roughness from image data has been developed, the computation of a 3-D fractal-based representation from actual image data has been demonstrated, and substantial progress has been made in the areas of shape-from-texture and texture segmentation. Characterization of image texture by means of a fractal surface model has also shed considerable light on the physical basis for several of the texture-partitioning techniques currently in use and has made it possible to describe image texture in a manner that is stable over transformations of scale and linear transforms of intensity.

In Appendix H [11], Pentland describes an interactive system for modeling natural forms. This system employes superquadrics, as well as fractal functions, in allowing the user simply and effectively to create and display almost any iconic object (e.g., the human form, surfaces with analytic descriptions, natural terrain, etc.).

This research is expected to contribute to the development of (1) a computational theory of vision applicable to natural surface shapes, (2) compact representations of shape useful for describing natural surfaces, and (3) real-time modeling, generation, and display of natural scenes. We also anticipate adding significantly to our understanding of the way humans perceive natural scenes.

B. Interactive Modeling and Analysis via Machine Synthesized Imagery

Terrain-Calc, described in Appendix I [12], is a system for synthesizing realistic sequences of perspective stereo views of real-world terrain (described within the machine by a database of geometric and photometric models). This system, implemented on a Symbolics 3600 Lisp Machine, has a sophisticated graphical interface, which allows the user to specify an arbitrary flight path over a modeled piece of terrain. A sequence of views (single images or stereo pairs, as desired), spaced at equal distances along the flight path, is generated at about one frame per minute, and up to 60 frames can be displayed at a rate of sixteen frames per second. This system is revolutionary in its flexibility, computational efficiency, and the quality of the renderings its produces, given that it does not employ any special-purpose hardware.

C. Architectures for Interactive and Real-Time Machine-Vision Systems

The computational demands imposed by interactive and real-time, machine-vision applications frequently exceed the capacity of conventional computer architectures. For this reason, attempts have been made to reduce computation time by decomposing serial algorithms into segments that can be simultaneously executed on parallel hardware architectures. Because many classes of algorithms do not readily decompose, one seeks some other basis for parallelism. In Appendix J [4] we show (1) that "guessing" the answer to a problem and then checking its validity is a useful approach and (2) that a number of vision algorithms are based on this concept. A parallel architecture capable of executing such algorithms is proposed.

V ACKNOWLEDGEMENT

The following researchers have contributed to the work described in this report: H. Baker, S. Barnard, R.C. Bolles, M.A. Fischler, P. Fua, M.J. Hannah, A.J. Hanson, D.L. Kashtan, K. Laws, O. Firschein, A.P. Pentland, L.H. Quam, G.B. Smith, T. Strat, and H.C. Wolf.

VI REFERENCES

[1] Barnard, S.T., "An Inductive Approach to Figural Perception," AIC Technical Note 325, SRI International, Menlo Park, California (September 1984); also Appendix F of this report.

[2] Bolles, R.C. and H.H. Baker, "Epipolar-Plane Image Analysis: A Technique for Analyzing Motion Sequences," to appear in the *Third International Symposium on Robotics Research,* Paris, France (October 1985); also Appendix E of this report.

[3] Fischler, M.A. and R.C. Bolles "Perceptual Organization and Curve Partitioning," *Proceedings of the 1983 Image Understanding Workshop* (June 1983); also IEEE CVPR-83.

[4] Fischler, M.A. and O. Firschein, "Parallel Guessing: A Strategy for High-Speed Computation," AIC Technical Note 338, SRI International, Menlo Park, California (September 1984); also Appendix J of this report.

[5] Fua, P.V., and A.J. Hanson, "Object Labeling Using Generic Knowledge," Appendix G of this report.

[6] Hannah, M.J., "Evaluation of STEREOSYS vs. Other Stereo Systems," Appendix A of this report.

[7] Hannah, M.J., "The Stereo Challenge Data Base," Appendix A of this report.

[8] Laws, K.I., "Goal-Directed Textured-Image Segmentation," Technical Note 334, SRI International (September 1984).

[9] Pentland, A.P., "Fractal Based Description of Natural Scenes," *Proceedings of the 1983 Image Understanding Workshop* (June 1983); and also IEEE CVPR-83.

[10] Pentland, A.P., "A New Sense for Depth of Field," *Proceedings of International Joint Conference on Artificial Intelligence,* Los Angeles, California (August 1985); also Appendix D of this report.

[11] Pentland, A.P., "Perceptual Organization and the Representation of Natural Form," AIC Technical Note 357, SRI International, Menlo Park, California (July 1985); also Appendix H of this report.

[12] Quam, L.H., "The Terrain-Calc System," Appendix I of this report.

[13] Smith, G.B., "Stereo Reconstruction of Scene Depth," *Proceedings of Computer Vision and Pattern Recognition '85,* San Francisco, California (June 19-23, 1985); also Appendix B of this report.

[14] Strat, T.M. and M.A. Fischler, "One-Eyed Stereo: A General Approach to Modeling 3-D Scene Geometry," *Proceedings of International Joint Conference on Artificial Intelligence,* Los Angeles, California (August 1985); also Appendix C of this report.

Appendix A

Evaluation of STEREOSYS vs. Other Stereo Systems

By: Marsha Jo Hannah

The Stereo Challenge Data Base

By: Marsha Jo Hannah

Evaluation of STEREOSYS vs. Other Stereo Systems

Marsha Jo Hannah

Artificial Intelligence Center, SRI International
333 Ravenswood Ave, Menlo Park, CA 94025

1 Introduction

As previously reported [Fischler, 1984], SRI International is implementing a complete, state-of-the-art stereo system that will produce dense three-dimensional (3-D) data from stereo pairs of intensity images. This system forms a framework for much of our stereo research and will be a base component of our planned expert system for 3-D compilation.

Ideally, we would assess the capabilities of our system by running it on a data set that has known ground truth against which to compare our results. Unfortunately, such data sets do not currently exist because of the extremely high cost of the ground work necessary to measure terrain elevations accurately for a close spacing and to assess the heights of all vegetation and buildings in the area. Lacking such a data set, we can only compare our results against those produced by other stereo systems or against the perceptions of a human looking at the same imagery in stereo on a CRT.

To test our system, currently called STEREOSYS, we have run it on several data sets, including two for which we also have results produced by the DIMP system at the U.S. Army Engineer Topographic Laboratories (ETL). Comparing our matching results to DIMP results or to human perception of what the correct match should be, we have begun to assess the strengths and weaknesses of STEREOSYS's matching techniques, as well as accumulating a catalog of examples of difficult areas for matching [Hannah, 1985]. A description of the experiments that we have conducted and our preliminary conclusions regarding the accuracy of the system are set forth in this report.

2 Description of Systems and Experiments

Our experiments compare the results of our automatic stereo system, STEREOSYS, against the results of the interactive DIMP system at ETL and against the stereo perceptions of an amateur photogrammetrist (the author of STEREOSYS [Hannah, 1984]). These systems and the design of our experiments are briefly described here.

2.1 Description of STEREOSYS

STEREOSYS (an improved version of STSYS, which was described in more detail in Hannah [1984]) is an automatic system for deriving disparity data, hence three-dimensional information, from a pair of aerial images of a scene, taken from moderately different points of view. This system operates on a hierarchy of different resolution versions of the pair of images, using normalized area correlation as its measure of whether areas in the two images are matched, that is, whether they represent the same point in space. STEREOSYS confines its attentions to image points having high information—the "interesting" points, which tend to be randomly spaced—and operates in several stages, using results from previous stages to constrain the search for points to be filled in at later stages. Overall, the system is very conservative about what constitutes a valid match; it will reject a questionable point at early stages of the processing (possibly filling it in later) in an attempt to produce the most reliable results possible.

2.2 Description of DIMP

The DIMP system, a descendant of the work of Panton [1978] and described in more detail in Norvelle [1981], is an interactively controlled system for deriving disparities. It operates on a single pair of high-resolution images, using normalized area correlation with the areas warped to take the slope of the terrain and the viewing geometry into account. DIMP finds a match for each point on a specified grid within the image, operating in raster scan fashion, with the expected disparity and terrain slope at a point predicted from the matches found at adjoining grid points in the preceding row and column. This system must be initialized manually. Because DIMP must record a match for each grid point (regardless of whether a match exists), and because it uses previous results (regardless of validity) to predict the next match, DIMP has a tendency to get off track, particularly in areas of low or ambiguous information, at places where the elevation or ground slope changes rapidly, or around artifacts in the images. For this reason, DIMP is manually coached—a human monitors its results constantly, interrupting the processing to get DIMP back on track as needed.

2.3 Description of Experiments

Comparing DIMP's grid-based results using warped correlation windows to STEREOSYS's randomly scattered results using ordinary correlation windows is a little like comparing apples and oranges. Matters are further complicated by the fact that, because of the noise properties of the images, STEREOSYS produced its best results in the 1024 × 1024 versions of these data sets, while DIMP used the 2048 × 2048 versions. However, we compared them in the following manner.

Comparisons were made only for those points for which STEREOSYS recorded an answer and were done at the resolution of the image in which STEREOSYS had operated. Points were said to have the same answer if the STEREOSYS result and the result at the closest DIMP grid point (scaled into the 1024 × 1024 image) were within one pixel of having the same disparity. Points about which there was disagreement were examined manually. The operator looked at both results, overlaid on the images at a variety of resolutions, both monocularly and using a stereoscopic viewer. The operator then decided which algorithm appeared to be in error and, based on experience with correlation algorithms, attempted to determine why the mistake had been made.

For data sets with no DIMP results, a much smaller number of points were matched. These were then compared with the human viewer's perception of what were the correct matches. Only the more blatant mistakes were detected and further analyzed.

3 Evaluations

In this section, we evaluate the performance of STEREOSYS on some of the data sets described in Hannah [1985]. For the first two sets, we have statistics as compared to the DIMP results; for the remaining sets, we give only general impressions of the results as seen by a human viewer.

3.1 The Phoenix Data Set

On the Phoenix data set, STEREOSYS found 5545 "interesting points," of which it thought it could reliably match 4676. Of these, only 43 disagreed significantly with the DIMP results for nearby points. Closer examination showed 15 of these to be uncorrected DIMP errors, 15 were STEREOSYS errors, 5 were points on which both systems appear to have made errors, and 8 were points for which the operator could not determine which system was in error. In most of the cases, the DIMP errors seemed to result from its algorithm having drifted gradually off track (usually starting in an area with little information), and its operator not catching it soon enough; the STEREOSYS approach of first providing a context in which to work, so that the code interpolates disparities, instead of extrapolating them, should remedy this problem. Most of the STEREOSYS errors (and almost all of the points for which the operator could not determine which algorithm was at fault) appeared to have resulted from an inappropriate threshold on the interest value: STEREOSYS was trying to match areas in which there was not enough information to make reliable matches. (The code has since been modified to be more selective about what it uses for "interesting" points.) Some of the STEREOSYS errors were due to not using warped correlation windows to account for the slopes. Most of the information in a window was in a corner of the window, so the disparity that was calculated was that of the corner, not the center of the window; using warped correlation or exponentially weighted correlation windows [Quam, 1984] would solve this problem. A fair number of the mistakes (particularly the ones in which both systems arrived at different wrong answers) were because of artifacts in the data—film grain, scratches, lint, hairs, fiducial marks, and the like; we are a long way from being able to understand, let alone automate, the human ability to identify offending objects and then ignore them in processing stereo data.

3.2 The Canadian Border Data Set

On the Canadian Border data set, STEREOSYS found 1428 "interesting points" (using a more restrictive threshold on interestingness), of which it decided it could reliably match 1262. Of these, 71 disagreed significantly with the DIMP results for nearby points, but only the 27 most blatant disagreements were examined by the operator. Close examination showed 9 of these to be uncorrected DIMP errors, 3 were STEREOSYS errors, 2 were points for which both systems appear to have made errors, and 13 were points for which the operator could not determine which system was in error. The reasons for the errors were highly varied. Most of the cases in which the operator was unable to fix the blame were forested portions of the image: the tree crowns looked sufficiently different in the two views that a naive human operator was unable to determine the correct match based purely on local context. In the face of this unmatchable data, DIMP had its usual trouble staying on track, particularly since this data set included a lot of discontinuities in depth between trees and ground, which DIMP's surface extrapolation algorithm is not designed to handle. STEREOSYS's errors happened around artifacts in the images, around the depth discontinuity at an overpass, and in an area of trees for which the true match was a subpeak on the correlation function.

3.3 The Moffett-Ames Data Set

Results on the Moffett set were somewhat limited by the lack of detailed camera calibration information to go with this imagery. STEREOSYS has been tuned to depend on having an accurate epipolar line for each point when matching points at later stages in the processing. Unfortunately, the crude relative camera model, which we were able to derive from the first few hierarchically matched points, proved to have significant errors as processing moved away from the center of the image. This meant that, for many points, the search for a match was started out quite far from the true match and frequently did not look far enough: many points failed to match at all, and several locked onto false matches that looked somewhat similar in the clutter of a suburban landscape. Because STEREOSYS was intended for use in a mapping scenario in which accurate camera information is the rule, no attempt has been made to modify it to work more reliably in the absence of accurate camera information.

3.4 The Lexington Reservoir Data Set

The Lexington data set was digitized for another project, which researched algorithms for handling raised objects. Because STEREOSYS is a conventional correlation system, it would not be expected to do well in the presence of depth discontinuities. As predicted, STEREOSYS coped well with the low features in the image and with the shadows of raised objects on the ground, but in areas containing discontinuities, it was unable to find matches that met its criteria for acceptance.

3.5 The Seattle I-5 Data Set

The I-5 data set is a prime example of the type of data on which edge matching triumphs over area matching. The information in the images is almost entirely straight lines resulting from the edges and lanes of the freeway. Most of the places that the "interest" statistic found to be suitable for correlation tended to be either false intersections (where one roadway crossed over another) or cars on the freeway, neither of which had matches in the second image. We were not able to get enough good matches to form even a crude camera model, so were unable to proceed with the processing.

3.6 The International Building Data Set

Despite the fact that STEREOSYS was designed for use on aerial photography, we have tried it on several ground-level pairs of images, just to get a feel for its limitations. We were pleasantly surprised to find that it did relatively well on the International Building set. Most of the interesting points were on the foreground plants; STEREOSYS coped quite well with these, probably because the background behind them was relatively uniform, so the discontinuities in depth did not cause the appearance of the correlation areas to change much. The only difficulties appeared to be with some false intersections, where two lines that seemed to meet in the image were actually separated in space.

3.7 The Machine Data Set

STEREOSYS also did fairly well with the Machine set. For the most part, it picked out the corners of various things on the machine and seems to have located plausible matches for most of the points it decided that it had matched correctly. There are a couple of questionable matches because of false intersections, which are unavoidable with an area-based matcher.

3.8 The Back Lot Data Set

STEREOSYS also did surprisingly well on the Back Lot data set. Of the points that it decided were well matched, only two were blatantly wrong—a car that is obscured in the second view looks quite similar to the car next to it, so the two interesting points on it are incorrectly matched. A few interesting points that the operator thought should have been easy to match were missed, probably because of interference in the hierarchical matching approach between the disparity of the near-field buildings and that of the background.

4 Conclusions

Our objective in constructing STEREOSYS was to implement a state-of-the-art, area-based system for stereo compilation operating on aerial photography. Along the way, we hoped to remedy some of the obvious problems we had seen with existing systems, such as DIMP's tendency to extrapolate itself off track. In this we have succeeded.

Because STEREOSYS uses fairly independent judgment on each match, it tends to avoid the problems we have seen in the DIMP results; indeed, on the Phoenix data set (and to a lesser degree on the Canadian Border data set), STEREOSYS was able to duplicate DIMP's correct results (for the points tried) and rectify a number of DIMP's mistakes. Although it happens rarely, it is still possible for STEREOSYS to make mistakes in the early stages of its processing, then propagate these mistakes into later matches. To avoid this, more work needs to be done on algorithms for detecting improperly matched points, so they can be removed before further processing.

The major criticism we have heard of STEREOSYS is that it produces matches at randomly spaced points (only where adequate information is present), when what is usually wanted is a closely spaced regular grid of elevation points, regardless of image content. So far, attempts at blindly interpolating the disparity data (ignoring the image data) as reported in Smith [1984] have proven less than satisfying. Marriage of the STEREOSYS techniques with something like DIMP, or with hierarchical warp correlation [Quam, 1984], or with image intensity based interpolation [Smith, 1985] or [Baker, 1982] might be profitable.

We have performed one experiment as a preliminary study in how to integrate the strengths of STEREOSYS with those of an edge-based matcher. The results of STEREO-

SYS were used as seeds for an edge-based matching system [Baker, 1982], which used the connectivity constraints of zero-crossing contours to control match propagation and which then did one iteration of its normal matching process. Because determining disparity constraints is a large part of the edge-based matcher's processing, introducing this information from STEREOSYS's results produced a significant improvement in the runtime of the edge-based matcher. The number of matched points increased by about an order of magnitude over the results of STEREOSYS alone. Although we have not yet finished a quantitative evaluation of these match accuracies, a qualitative analysis indicates that the results from the combined technique are significantly more accurate than the results of the edge-based system alone.

Overall, we have found that STEREOSYS performs credibly on the low-resolution aerial imagery for which it was designed. It has difficulties when processing areas that violate its premises about the continuity of the world, but linking it with an edge-based matcher (which would excel in these types of areas) seems to be a promising approach.

Acknowledgements

The research reported herein was supported by the Defense Advanced Research Projects Agency under Contract MDA903-83-C-0027, which is monitored by the U.S. Army Engineer Topographic Laboratory. The views and conclusions contained in this paper are those of the author and should not be interpreted as necessarily representing the official policies, either expressed or implied, of the Defense Advanced Research Projects Agency or of the United States Government.

I would like to thank Robert Bolles, Lynn Quam, Harlyn Baker, and Martin Fischler for their support on this project.

References

Baker, H. Harlyn, 1982. "Depth from Edge and Intensity Based Stereo," Ph.D. Thesis, Stanford University Computer Science Department Report STAN-CS-82-930, September 1982.

Fischler, Martin A., 1984. *Computer Vision Research and Its Applications to Automated Cartography: (Combined) Second and Third Semiannual Technical Reports,* SRI International, Menlo Park, CA, September, 1984.

Hannah, Marsha Jo, 1984. "Description of SRI's Baseline Stereo System", in Fischler [1984].

Hannah, Marsha Jo, 1985. "The Stereo Challenge Data Base", SRI International Artificial Intelligence Center Technical Memo, in preparation, August, 1985.

Norvelle, F. Raye, 1981. "Interactive Digital Correlation Techniques for Automatic Compilation of Elevation Data," U.S. Army Engineer Topographic Laboratories Report ETL-0272, October, 1981.

Panton, Dale J., 1978. "A Flexible Approach to Digital Stereo Mapping," *Photogrammetric Engineering and Remote Sensing,* Vol. 44, No. 12, pp. 1499-1512.

Quam, Lynn H., 1984. "Hierarchical Warp Stereo," in Fischler [1984].

Smith, Grahame B., 1984. "A Fast Surface Interpolation Technique," in Fischler [1984].

Smith, Grahame B., 1985. "Stereo Reconstruction of Scene Depth," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition,* San Francisco, CA, June 9-13, 1985.

The Stereo Challenge Data Base

Marsha Jo Hannah

Artificial Intelligence Center, SRI International
333 Ravenswood Ave, Menlo Park, CA 94025

1 Introduction

As previously reported in Fischler [1984] and Hannah [1984], SRI International is implementing a complete, state-of-the-art stereo system that will produce dense three-dimensional (3-D) data from stereo pairs of intensity images. Ideally, we would assess the capabilities of our system by running it on a data set that has known ground truth against which to compare our results. Unfortunately, such data sets do not currently exist, because of the extremely high cost of the ground work necessary to measure terrain elevations accurately for a close spacing and to assess the heights of all vegetation and buildings in the area. Lacking such a data set, we can only compare our results against those produced by other stereo systems, or against the perceptions of a human looking at the same imagery in stereo on a CRT.

To test our system, currently called STEREOSYS, we have run it on several data sets, including two for which we also have results produced by the DIMP stereo system at the U.S. Army Engineer Topographic Laboratories. While comparing our matching results to DIMP results or to human perception of what the correct match should be, we have begun to accumulate a catalog of examples of difficult areas for stereo processing.

In this report, we describe several datr sets that we have processed and discuss the types of problems that our matching algorithms have encountered. This information is part of the "stereo challenge data base" we are assembling to test matching algorithms against; the actual data base will contain many more instances of hard-to-match places than are shown in the simple examples illustrated here.

2 Data Sets Processed by STEREOSYS

The following data sets have been processed through STEREOSYS, our stereo compilation program. The areas noted are examples of types of areas that STEREOSYS had incorrectly matched (as compared with other computer algorithms or with human stereo results), ones that STEREOSYS was unable to match well enough to suit its internal criteria, or ones on which STEREOSYS was unable to do anything for lack of information in the imagery.

2.1 The Phoenix Data Set

Most of our area-based processing and analysis to date, as well as some edge-based processing, has been done on a data set that we received from the U.S. Army Engineer Topographic Laboratories (ETL). The imagery consists of a pair of 2048 × 2048 pixel images representing a 2" × 2" portion from two standard 9" × 9" mapping photographs taken over Phoenix South Mountain Park, near Phoenix, Arizona. The data covers approximately a 2-km square of high desert, both plain and steep hills, dotted with brush; the beginnings of an agricultural area is at one edge of the images.

This data set is known locally as the Phoenix set. In addition to the images, this data set also contains camera information in the form of absolute position and orientation data, internal calibrations for the camera, and rectification polynomials to account for the digitization process. We also have a set of results from the interactively coached DIMP stereo compilation system at ETL [Norvelle, 1981] in the form of an array of the matching points for a grid of image points (every 5th pixel) and the arrays of 3-D positions derived from these matched point pairs.

This data set provides a number of challenges to stereo processing algorithms, particularly to those based on area correlation. (Numbers in parentheses refer to the example points in Figure 1 and Table 1.) At least half of the terrain in the imagery is very steep (1), so that an area on the ground frequently projects to windows of different sizes and shapes in the two images; this frequently results in poor correlations or in mismatches. There are some portions of the terrain that have little vegetation, giving correlation algorithms insufficient or unreliable information with which to work (12). The agricultural area contains some very straight roads surrounded by land without distinguishing visual texture (2), causing matches to "slide" along the roads until the noise in the images matches best. Some of the roads contain cars that have moved in the time between the two images (3), rendering those areas difficult to match. The images also include portions of regularly spaced orchards (4, 5, 6), which can lead to local confusion by the matcher, because all the trees look alike and have very similar context. In the agricultural area, a few buildings (7) cause depth discontinuities that can be difficult for the matcher.

The Phoenix data set is made more challenging because the imagery is of somewhat poor quality, with scratches (8), pen marks (9), fiducial marks (10), hairs (11), and the like, which have been digitized into the data. The photographs also appear to have been digitized at the maximum possible resolution—the film grain (12) is apparent in otherwise low-information areas of the imagery, leading to random mismatches.

2.2 The Canadian Border Data Set

We have also done a significant amount of processing on a data set received from the Defense Mapping Agency (DMA). The imagery consists of a pair of 2048 × 2048 pixel images representing a portion of two mapping photographs taken somewhere along the U.S.-Canadian border. The data set covers an area of gently rolling terrain cut by a steep ravine and crossed by a major highway; the ground cover is a mixture of forested areas having sharp boundaries with areas that have been cleared for crop lands; the imagery also contains several farm complexes and a town.

This data set is known locally as the Canadian Border set, or, more simply, the Canada

Point	x	y	Description
1	1136	436	Steep ridge
2	1616	420	Ambiguity along road
3	1972	286	Car moved on road
4	1892	526	Regular pattern in orchard
5	1924	586	Horizontal ambiguity along orchard edge
6	1954	482	Vertical ambiguity along orchard edge
7	1950	722	Discontinuity at building
8	1178	140	Digitized scratch on photo
9	1502	636	Pen mark on photo
10	1236	862	Fiducial mark on photo
11	1726	170	Hair on photo
12	1642	912	Digitized film grain

Table 1: Examples from lower right quarter of Phoenix imagery

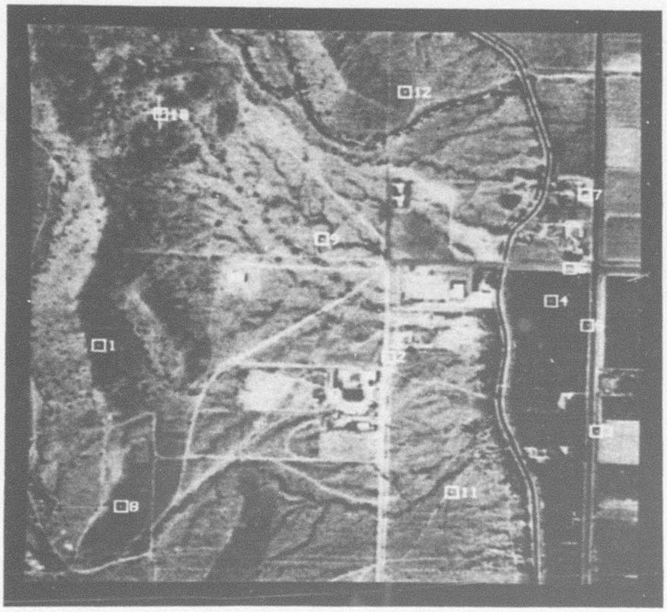


Figure 1: Lower left quarter of Phoenix image at 1024 × 1024 resolution

set. In addition to the images, this data set also contains camera information, in the form of absolute position and orientation data, internal calibrations for the camera, and rectification polynomials to account for the digitization process. We also have a set of results from the interactively coached DIMP stereo compilation system at ETL in the form of an array of the matching points for a grid of image points (every 10th pixel).

This data set is extremely challenging for stereo processing algorithms, whether based on area correlation or edge matching. (Numbers in parentheses refer to the example points in Figure 2 and Table 2.) The major problem encountered in these images is the tree cover. In some areas, the trees are very dense and in full foliage so that the ground cannot be seen at all (1, 2, 3). In other areas, the trees are more sparse so a particular window might contain both tree tops and ground, which match at different disparities (4); this also happens at the edge of a dense forest (5) and where a narrow row of trees lines a field (6). In many cases, the tree tops contain enough detail that they present a much different appearance in the two images making any sort of matching is a problem, let alone separating tree elevation from ground elevation. The steep terrain in the vicinity of the ravine compounds the problem, causing the vegetation to be foreshortened differently in the two views (7). There is a large building complex in the ravine, further complicating the matching problem by introducing partial occlusions along its walls (8). There is also a highway bridge over the ravine (9) and a highway overpass (10), both of which cause similar problems because of occlusions. Straight highways (11), with an occasional car that moved between the times of the two views, cause the usual problems, as do agricultural fields (12) with little internal visual information. As with the Phoenix set, film grain and various artifacts such as hairs, scratches (13), and pen marks (12) all have negative effects on matching algorithms.

2.3 The Moffett-Ames Data Set

We have also processed an urban data set received from the Defense Mapping Agency. The imagery consists of a pair of 1024 × 1024 pixel images representing a portion of two mapping photographs taken over the Moffett Field Naval Air Station and the NASA Ames Research Center including portions of the cities of Mountain View and Sunnyvale, California. The data covers an area of generally level terrain adjoining San Francisco Bay; in addition to the airfield and hangers, the area includes salt evaporator ponds, agricultural fields, housing developments, and office complexes and is crossed by a major highway.

This data set is known locally as the Moffett-Ames set or, or more simply, the Moffett set. This data set came with camera information (absolute position and orientation data, internal calibrations for the camera, and rectification polynomials to account for the digitization process), but we have been advised that this information contains errors, so have not attempted to use it. At present, we have no other matching results for this data set, although it is rumored that some form of ground truth exists.

This data set has a number of challenging features for stereo processing algorithms, whether based on area correlation or edge matching. (Numbers in parentheses refer to the example points in Figure 3 and Table 3.) Most of the features in the images are man-made structures of one form or another; this leads to strong linear edges along roads (1) and airfield runways (2), which are troublesome for area correlation. There are a number of large buildings in the area, including Moffett's blimp hanger (3), NASA's wind tunnel (4), and

Point	x	y	Description
1	698	752	Dense trees with dark foliage
2	334	1822	Dense trees with medium-intensity foliage
3	850	662	Dense trees with light foliage
4	396	444	Mixed trees and ground
5	808	862	Edge of dense trees
6	196	1606	Row of trees between fields
7	888	1632	Trees in ravine
8	2000	1182	Large buildings in ravine
9	968	1580	Highway bridge over ravine
10	1058	1208	Highway overpass
11	1592	794	Ambiguity along highway
12	1162	86	Pen marks in field
13	420	1992	Scratches on photo

Table 2: Examples from Canada imagery

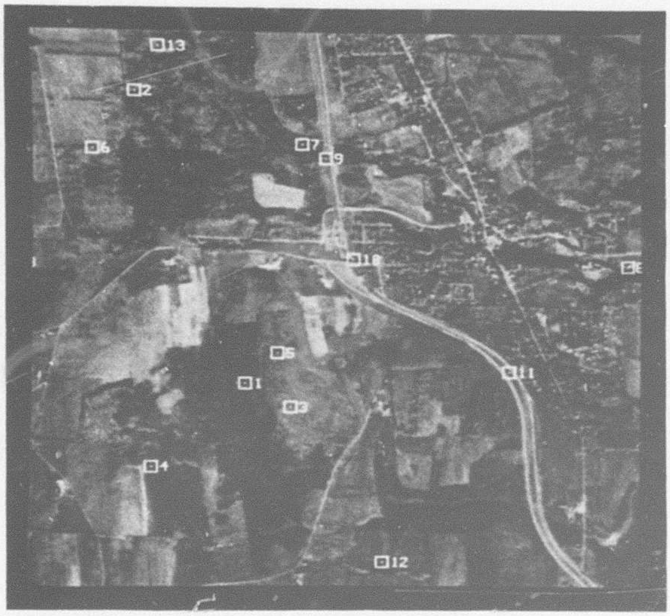


Figure 2: Canada image at 512 × 512 resolution

Point	x	y	Description
1	736	675	Edge of US-101
2	589	665	Edge of Moffett runway
3	338	662	One of Moffett's blimp hangers
4	463	322	NASA's wind tunnel
5	681	945	Lockheed's Building 001
6	438	206	Trailer park
7	628	438	Rows of barracks at the naval station
8	676	186	Similar blocks of regularly spaced houses
9	881	855	Rows of identical light industrial buildings
10	557	935	Parking lots with regular patterns of cars
11	677	735	Agricultural fields
12	76	760	Salt ponds
13	186	838	Specular reflection on salt pond
14	238	909	Specular reflection on salt pond

Table 3: Examples from Moffett imagery

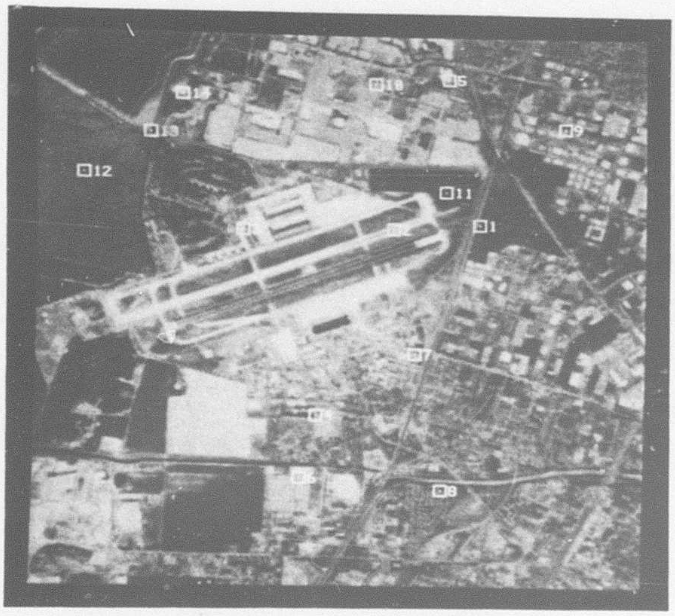


Figure 3: Moffett image at 512 × 512 resolution

Lockheed's Building 001 (5), which present the usual problems with partial occlusions. The imagery includes a variety of suburban housing, whose fine detail will be difficult for edge matching algorithms to handle. In addition, there are several repetitive patterns in these images, such as rows of trailers in a trailer park (6), rows of barracks at the naval station (7), blocks of regularly spaced houses (8), rows of identical light industrial buildings (9), and parking lots with regular patterns of cars (10). There are the usual problems with large blank areas such as the agricultural fields (11) and the salt ponds (12). The salt ponds are particularly troublesome, because the motion of the camera along the flight path causes some of these ponds (13, 14) to show specular reflections in one image, but not in the other; this causes contrast reversals with the surrounding dams, which will confound most area and edge matchers.

On the positive side, this data set appears to be relatively clean; that is, it is free from the scratches, lint, hairs, pen marks, and other artifacts that frequently compound the problem with aerial imagery. However, the lack of precise camera information severely handicapped our processing of this imagery, because the images appear to have a significant distortion near their edges. The crude relative camera model calculated from the first few matched points was significantly in error (i.e., human-indicated matching points were several pixels away from the predicted epipolar lines) over much of the image; this resulted in many points which failed to match at all, as well as a number of falsely accepted mismatches, because of the ambiguities inherent in urban scenes.

2.4 The Lexington Reservoir Data Set

We have partially processed a data set that we digitized ourselves from aerial images received from the Defense Mapping Agency. The imagery consists of a pair of 512 × 512 pixel images representing a small portion of two mapping photographs taken along Highway 17 in the vicinity of Lexington Reservoir near Los Gatos, California. The data is a high-resolution view of a relatively small area, including a part of the freeway, a small water storage tank, part of a large tank, a small building, a few trees, and a hill.

This data set is known locally as the Lexington Reservoir set or, more simply, the Lexington set. We do not have camera information for this data set, nor do we have other matching results for it.

This data set provides a severe challenge for ordinary matching algorithms. (Numbers in parentheses refer to the example points in Figure 4 and Table 4.) Large areas of the data have no visual information, such as the concrete aprons around the tanks (1), asphalt service roads (2), or grassy hillsides (3). The tops of the trees (4, 5) are seen from much different perspectives and so have radically different appearances. The linear edges between the bland areas cause the usual problems, as does the highway itself (6); the car (7) that has moved between the two views also causes matching problems. Because the images are such high resolution, the discontinuities in the image around the small tank (8) and the building (9) are a significant problem. For the ultimate challenge, there is also an isolated power pole (10) to attempt to match.

On the positive side, this data set appears to be relatively free from the scratches, lint, and other artifacts that frequently compound the problem with aerial imagery. However, the high resolution was obtained by digitizing down to the film grain, so many of the "features" found by the interest operator are really noise in otherwise blank areas.

Point	x	y	Description
1	496	366	Concrete apron around a tank
2	410	404	Asphalt service road
3	228	162	Grassy hillside
4	126	258	Tree top
5	42	348	Tree top
6	178	28	Highway 17
7	80	38	Car that has moved between the two views
8	230	272	Small tank, up on stilts
9	396	320	Building
10	384	146	Power pole

Table 4: Examples from Lexington imagery

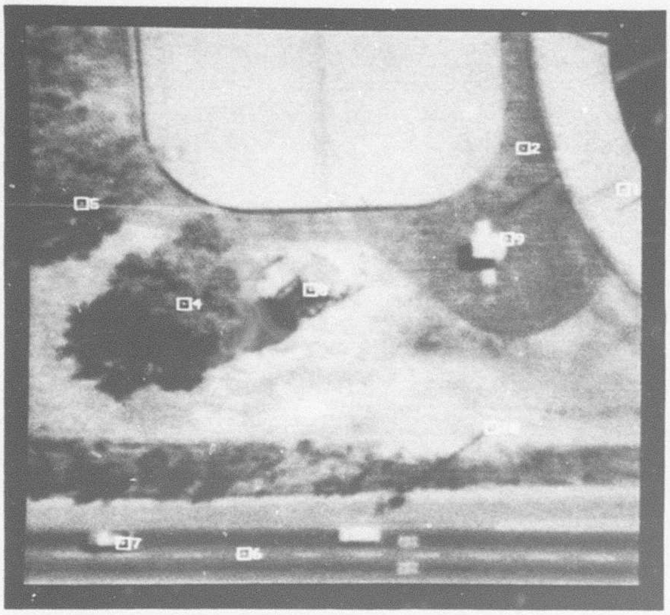


Figure 4: Lexington image at 512 × 512 resolution

2.5 The Seattle I-5 Data Set

We have partially processed a data set acquired from Boeing. The imagery consists of a pair of 200 × 200 pixel images from mapping photographs taken over the interchange of Interstate 5 with Spokane Street in Seattle, Washington. The data is a medium-resolution view of a relatively small area, featuring part of this major freeway interchange.

This data set is known locally as the Seattle I-5 set or, more simply, the I-5 set. We do not have camera information for this data set, nor do we have matching results other than those area- and edge-based matches we have produced on it.

This data set provides many good features for edge matching, but a severe challenge for area-based matching algorithms. (Numbers in parentheses refer to the example points in Figure 5 and Table 5.) The vast majority of the information in the images lies along the various roadways, both in their external edges (1) and in the internal edges between lanes (2). Our "interest" operator will not select areas containing only linear structures, but readily selects places where one linear structure intersects another. Unfortunately, such points occur mainly where one roadway crosses over another (3, 4). Because these are not true intersections (i.e., the freeway and its overcrossing do not actually intersect, but merely appear to do so in most views), such points rarely have a proper match in a different view of the scene. Unfortunately, they do have very well-correlated false matches, which occur where the two linearly-ambiguous structures falsely intersect in the second photo. Also highly "interesting" are points where the linear pattern of the road is obscured by a car (5), which, of course, has a different position in the other image. In addition to the problems of obscuration caused by the discontinuities between the levels of the roadway (6), there are also the usual pr-blems with foreshortening on the steep banks leading from one level of the interchange to another (7) and with the relatively blank areas of landscaping in some of the adjoining areas (8).

As presently implemented, our stereo system was unable to do much with these images. So many of the points were either unmatchable or had false matches that we were unable to obtain even a crude relative camera model for these images; hence, we were unable to proceed. An edge-matching algorithm, started with carefully hand-picked initial matching points, was able to derive the model it needed and process most of the image, although it had difficulties with the ambiguities inherent in the similar, parallel lanes of the freeway.

2.6 The International Building Data Set

We have also processed several ground-level stereo data sets digitized locally from pictures taken with a hand-held 35-mm camera. The first of these sets consists of a pair of 450 × 450 pixel images taken in the patio of the International Building at SRI in Menlo Park, California. In the foreground are three large pots containing a small tree, a bush, and some succulents; in the background are a few chairs in front of a wall of the building.

This data set is known locally as the International Building set. We do not have camera information for this data set, nor do we have matching results other than those we have produced on it.

This data set provides some very interesting challenges for all types of matching algorithms. (Numbers in parentheses refer to the example points in Figure 6 and Table 6.) The little tree in the foreground (1) is quite diffuse, so almost any window within the tree

Point	x	y	Description
1	46	158	Edge of I-5
2	13	68	Edges between lanes of I-5
3	65	128	Pseudo-intersection of two roadways
4	114	95	Pseudo-intersection of two roadways
5	24	144	Car which moved between images
6	138	82	Discontinuities between levels of the roadway
7	190	141	Foreshortening on steep banks
8	31	56	Featureless areas of landscaping

Table 5: Examples from I-5 imagery

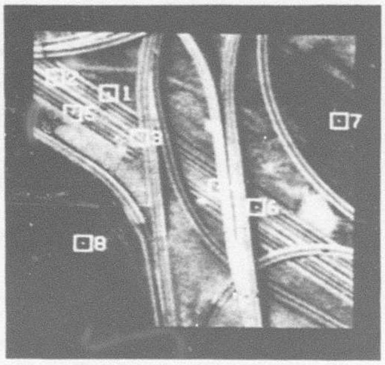


Figure 5: I-5 image at 200 × 200 resolution

Point	x	y	Description
1	288	275	Diffuse foreground tree
2	226	286	Background behind tree
3	80	336	Reflection in window
4	39	196	Near-field occlusions
5	425	203	Pseudo-intersections
6	375	205	Linear column edge
7	104	419	Blank ceiling

Table 6: Examples from International Building imagery

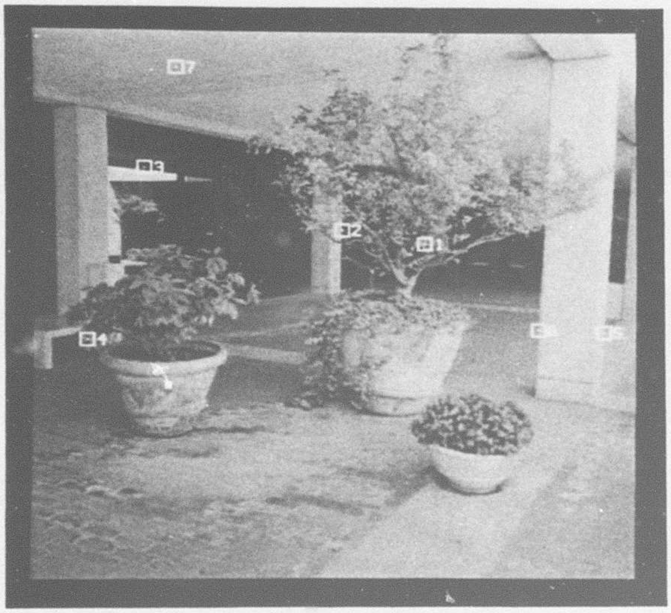


Figure 6: International Building image at 450 × 450 resolution

will also contain pixels from the background (2); the trick is to separate them. The large windows in the middle ground (3) contain very clear reflections of objects out of the field of view of the images; these objects are matchable, but will receive spurious depths, because the depth triangulation calculations assume that lines of sight are straight. Extreme near-field objects will cause the usual problem with occlusions (4) and pseudo-intersections (5). Of course, area-based measures will have their usual difficulties with linear features such as the columns (6) and blank areas such as the ceiling (7).

2.7 The Machine Data Set

Another of the ground-level stereo data sets we have processed was also digitized locally from pictures taken with a hand-held 35-mm camera. This set consists of a pair of 500 × 500 pixel images taken in one of the parking lots at SRI in Menlo Park, California. In the foreground is a large piece of machinery (probably a diesel-powered generator) sitting on blocks, and behind it is an oblique view of a building with a few small trees planted along it and part of a row of cars parked in front of it.

This data set is known locally as the Machine set. We do not have camera information for this data set, nor do we have matching results other than those we have produced on it.

This data set provides some interesting challenges for matching algorithms. (Numbers in parentheses refer to the example points in Figure 7 and Table 7.) The radiator of the machine (1) is seen at a rather oblique angle, so is foreshortened differently in the two views; the digitization also brought out interesting moire patterns, which differ in the two views. The electric truck behind the machine (2) has been driven away between the times of two views, complicating matches in that area. The exhaust stacks on the machine (3) create pseudo-intersections with the building, which will cause difficulties for most matchers. The car fender (4) is occluded by the machine in the second view. The machine contains a great deal of fine detail, such as wiring (5), whose narrowness presents problems for the matcher. Much of the detail on the building (6) is linear and very nearly parallel with the epipolar line, so is difficult for area- or edge-based matchers to handle properly. The building itself (7) and the asphalt of the parking lot (8) both contain little information, with just enough noise introduced by the digitization to cause trouble.

2.8 The Back Lot Data Set

Another of the low-angle stereo data sets we have processed was also digitized locally from pictures taken with a hand-held 35-mm camera. This set consists of a pair of 254 × 254 pixel images taken from the roof of one of the buildings at SRI in Menlo Park, California. The scene is framed by two large buildings at each side of the imagery; seen between the buildings are two rows of cars parked along a street with a low building behind them and lots of trees behind that.

This data set is known locally as the Back Lot set, or more simply, the Lot set. We do not have camera information for this data set, nor do we have matching results other than those we have produced on it.

This data set provides some interesting challenges for matching algorithms. (Numbers in parentheses refer to the example points in Figure 8 and Table 8.) The most difficult problem posed by this data set is how to deal with points that are unmatchable, because

Point	x	y	Description
1	139	288	Radiator foreshortened, with moire pattern
2	88	289	Truck moves between frames
3	216	397	Exhaust stack pseudo-intersects building
4	106	336	Fender occluded
5	324	245	Wiring detail on machine
6	457	421	Linear feature, paralleling epipolar lines
7	168	435	Blank wall
8	80	88	Blank pavement

Table 7: Examples from Machine imagery

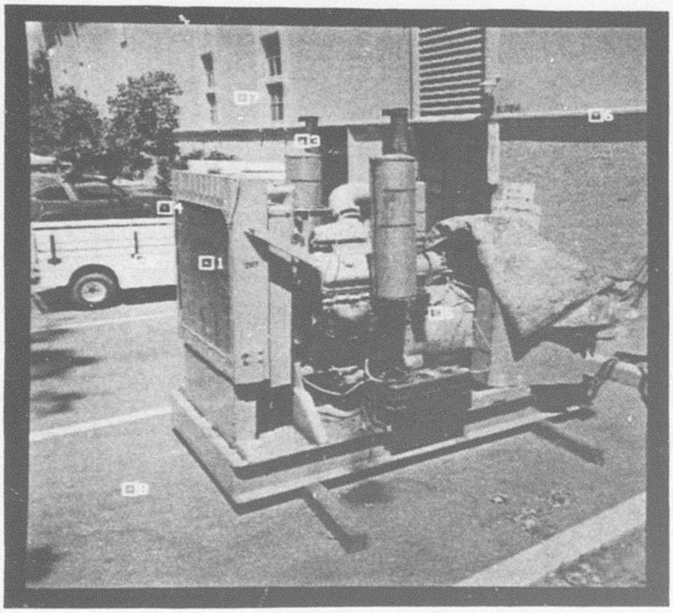


Figure 7: Machine image at 500 × 500 resolution

Point	x	y	Description
1	172	32	Front wheel of car obscured in 2nd image
2	170	91	Car obscured in 2nd image
3	91	62	Cars foreshortened differently
4	124	224	Tree structure ambiguous
5	168	227	Tree nearly obscured
6	183	76	Linear building edge
7	157	118	Linear roof line, paralleling epipolar lines
8	221	64	Blank wall
9	101	40	Blank ground

Table 8: Examples from Back Lot imagery

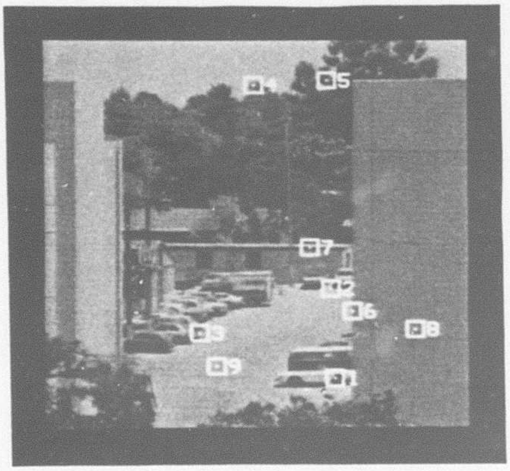


Figure 8: Back Lot image at 254 × 254 resolution

of occlusions. The strip of data just to the left of the edge of the right-hand building does not appear in the second image, because of the change in point of view. This means that the front wheel of the first car in that row (1) and the partially visible car in the back row (2) do not have valid matches in the second image, but a window containing the front wheel of the first car (1) looks quite like a window containing the back wheel of that car, leading to a mismatch with a fairly good correlation; similarly, the car in the back (2) looks enough like the car next to it to cause a persistent mismatch. The cars in the other row (3) are foreshortened or occluded just enough to make matching difficult. The humps and bumps in the skyline tree edge (4) are sufficiently similar to cause mismatches. Hierarchical techniques did not work well on the tree (5) behind the building at the right, seeming to lock onto the building corner instead of the tree in the low resolution versions of the image. There were the usual problems with linear edges (6), especially the ones parallel to the epipolar lines (7), as well as problems with areas that had marginal information, such as the buildings (8) and the parking lot (9).

3 Other Data Sets

We have available several more data sets that we have not processed as yet. From our experience, however, we feel that each of these data sets provides some interesting challenges for stereo processing. We note these in passing.

3.1 The Washington Monument Data Set

We have a pair of 512 × 512 pixel images acquired from Carnegie-Mellon University; these were taken over the Washington Monument in Washington, DC (see Figure 9). This is a fairly wide-angle pair so that many of the buildings have one vertical face shown in one image and the opposing face shown in the other; these occlusions will significantly complicate matching. A fair amount of traffic on the streets has moved in the time between the two images. The strong linear patterns of the streets and the blank roof tops will cause the usual problems for area-matching algorithms; the detail on some of the building sides may confuse edge-based methods.

3.2 The Fort Belvoir Doublet Data Set

We have a pair of 512 × 512 pixel images received from the Defense Mapping Agency; these were taken near Fort Belvoir, Virginia (see Figure 10). The images show part of a freeway with the usual moving traffic as well as a petroleum tank farm. Because this is a fairly wide-angle pair, the amount of visible tank face varies between the images. In a number of areas, the trees have apparently shed their leaves for the winter, as the shadows of the trunks are visible on the ground through a "haze" of upper branches—a difficult situation for area- and edge-based matchers alike. The images are "contaminated" with a large black triangle, which was apparently drawn on the original photograph before it was digitized. Camera information is reputed to be available for these images, but is rumored to contain errors.

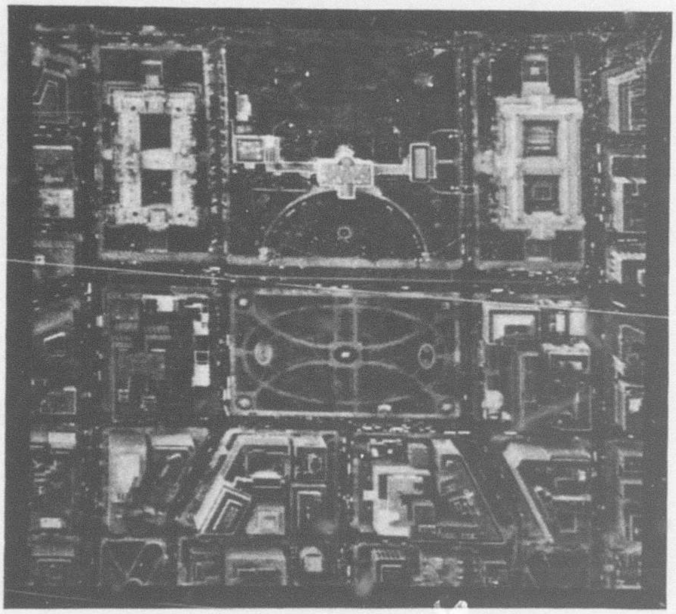


Figure 9: Washington Monument image at 512 × 512 resolution

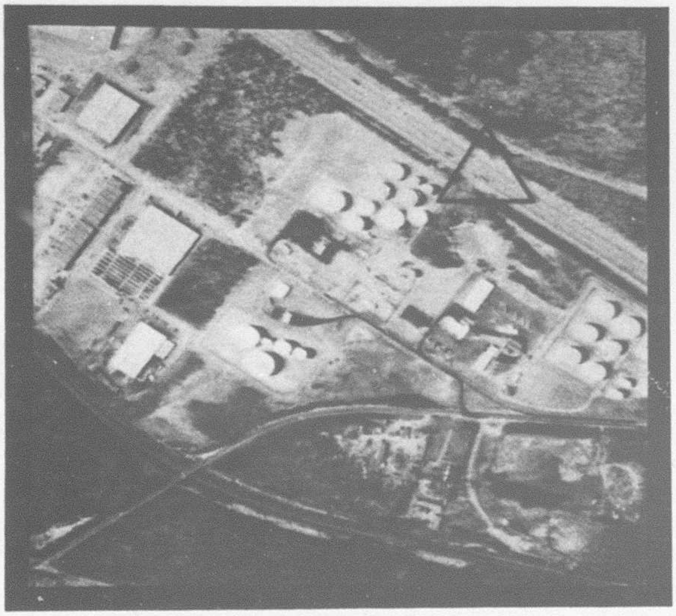


Figure 10: Fort Belvoir Doublet image at 512 × 512 resolution

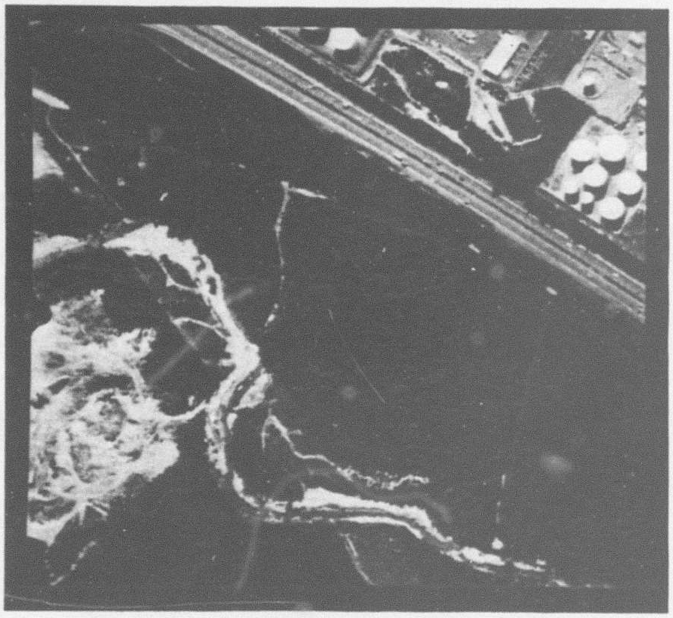


Figure 11: Fort Belvoir Triplet image at 512 × 512 resolution

3.3 The Fort Belvoir Triplet Data Set

We also have a trio of 512 × 512 pixel images received from the Defense Mapping Agency; these were taken near Fort Belvoir, Virginia (see Figure 11). The images show part of a freeway with the usual moving traffic, as well as a large area of forest, a steep ravine, a gravel quarry, and what appears to be an office complex under construction; a portion of the petroleum tank farm featured in the Fort Belvoir Doublet also appears in a corner of some of the images. Most of the area of the images is covered with trees, which are in full leaf; the crowns provide a relatively bland area with detail differing greatly in the two views. An interesting challenge is matching the high-tension power transmission towers, which appear at various places across the images. The images are "contaminated" with some of the edge markings on the original photographs, because the edges were not clipped before digitization. Also, the contrast and brightness of the images is not constant—the third image differs significantly from the other two, which may confound some matching algorithms. Camera information is reputed to be available for these images, but is rumored to contain errors.

3.4 The Phone Data Set

We also have a pair of 256 × 256 pixel images of a telephone sitting on a desk top (see Figure 12), which forms quite a challenge for stereo processing. On the desk, in addition to the phone, there is a decorated porcelain coffee mug containing a pencil. The background behind the scene is slightly out of focus and contains a sparse, but highly ambiguous pattern,

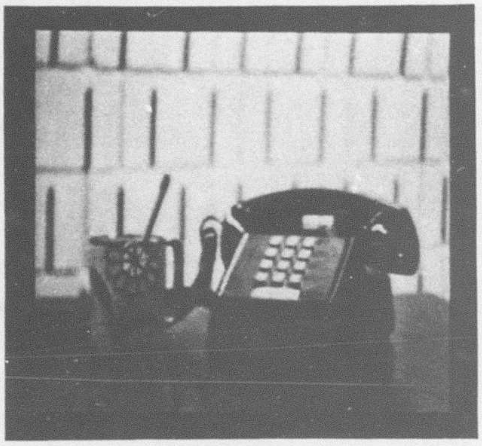


Figure 12: Phone image at 256 × 256 resolution

which most stereo algorithms match incorrectly. The change in point of view results in a significant rotation of the scene, so most of the objects are foreshortened differently between the two views.

3.5 The Chair Data Set

We also have a trio of 256 × 192 pixel images taken of two chairs (see Figure 13). The two chairs, one a secretarial swivel chair, the other a conference room stackable chair, each contain relatively little detail, and their background is a wall that is almost the same intensity as the chairs. Other objects in the scene include a chart of some type hanging askew on the wall, a large soft-drink cup on the secretarial chair, a small oscilloscope on the stackable chair, and a tablelike object in the foreground with two unidentified objects on it. Both of the chairs have reflections of the ceiling light fixtures on their vinyl coverings, and there is an artifact common to the 3 images in the lower left corner: a black corner with a white bar across it. The lack of features and the indistinct edges will make this a challenging data set for most stereo algorithms.

3.6 The Motion Data Sets

We also have available some motion sequences of images taken in the robotics laboratory, which had been cluttered with a variety of house plants and other objects to make the problem more interesting (Figure 14 shows a typical scene). These images were taken

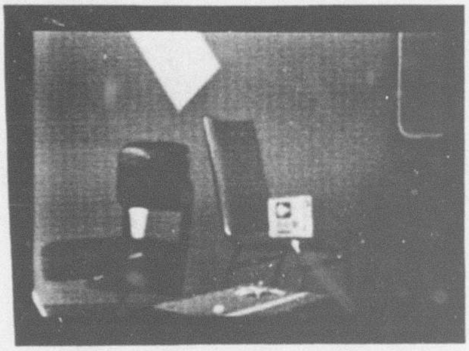


Figure 13: Chair image at 256 × 192 resolution

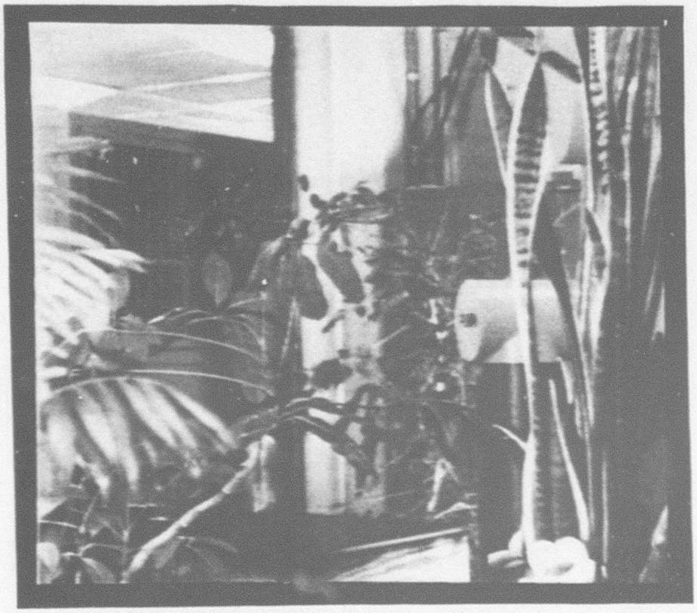


Figure 14: A typical Motion image at 490 × 480 resolution

with a CCD video camera mounted on an x-y table, which was moved in 125 steps of 0.2" each, in a straight line either laterally or forward; because the camera was precisely controlled, it should be possible to recover the camera information. All of the scenes are quite complicated, with near-field objects that change relative positions with respect to objects in the background from frame to frame, some areas of nearly constant intensity, and many pseudo-intersections, where edges that do not meet in the real world appear to intersect in the images. The large number of images (currently available on the LISP-Machines, but a few may be transferred to the VAX for more study) makes it possible to experiment with optic flow techniques, stereo at a variety of baseline lengths, stereo combined with motion, and the like.

Acknowledgements

The research reported herein was supported by the Defense Advanced Research Projects Agency under Contract MDA903-83-C-0027, which is monitored by the U.S. Army Engineer Topographic Laboratory. The views and conclusions contained in this paper are those of the author and should not be interpreted as necessarily representing the official policies, either expressed or implied, of the Defense Advanced Research Projects Agency or of the United States Government.

I would like to thank Robert Bolles, Lynn Quam, Harlyn Baker, and Martin Fischler for their support on this project.

References

Baker, H. Harlyn, 1982. "Depth from Edge and Intensity Based Stereo," Ph.D. Thesis, Stanford University Computer Science Department Report STAN-CS-82-930, September 1982.

Fischler, Martin A., 1984. *Computer Vision Research and Its Applications to Automated Cartography: (Combined) Second and Third Semiannual Technical Reports,* SRI International, Menlo Park, CA, September, 1984.

Hannah, Marsha Jo, 1984. "Description of SRI's Baseline Stereo System", in [Fischler, 1984].

Norvelle, F. Raye, 1981. "Interactive Digital Correlation Techniques for Automatic Compilation of Elevation Data," U.S. Army Engineer Topographic Laboratories Report ETL-0272, October, 1981.

Appendix B

Stereo Reconstruction of Scene Depth

By: Grahame B. Smith

Stereo Reconstruction of Scene Depth

Grahame B. Smith

Artificial Intelligence Center, SRI International
Menlo Park, California 94025

Abstract

The conventional approach to the recovery of scene topography from a stereo pair of images is based both on the identification of distinctive scene features and on the application of constraints imposed by the viewing geometry. We offer a new prescription for recovering a relative-depth map. We integrate image irradiance profiles to find dense relative-depth profiles. Our procedure neither matches image points (at least in the conventional sense) nor "fills in" data to obtain the dense depth map. Although there are outstanding problems associated with depth discontinuities and image noise, the technique is effective.

1. Introduction

The conventional approach to the recovery of scene topography from a stereo pair of images (or from a motion sequence) is based on the identification and matching of distinctive scene features and on the satisfaction of constraints imposed by the viewing geometry. Typically, three steps are required: determination of the relative orientation of the two images, computation of a sparse depth map, and derivation of a dense depth map for that scene.

In the first step, points corresponding to unmistakable scene features are identified in each of the images. The relative orientation of the two images is then calculated from these points. This is, in part, an unconstrained matching task. Corresponding image features must be found. Without a priori knowledge, such a matching procedure knows neither the approximate location (in the second image) of a feature found in the first image, nor the appearance of that feature. We may often assume that appearance will vary little between images and that they were taken from similar positions relative to the scene, but this assumption is based on a priori knowledge of the acquisition process.

Recovery of the relative orientation of the images reduces the computation of a sparse depth map from unconstrained two-dimensional matching to constrained one-dimensional matching. The hunt for a scene feature identified in the first image is reduced to a one-dimensional search along a line in the second image. Identification of this feature in the second image makes it possible to calculate disparity, and hence relative scene depth, for the feature.

Identification of corresponding points in the two images is based primarily on correlation techniques. Area-based correlation processes may be applied directly to the raw image irradiances or to images that have been preprocessed in some manner. For example, edges (identified by the zero crossings of the Laplacian of their image irradiances) have been used in obtaining correspondences.

The outcome of this second step is a sparse map of the scene's relative depth at those points that were identified in both images of the stereo pair.

A sparse depth map does not define the scene topography. The third and final step in recovering the topography of the scene is "filling in" this sparse map to obtain a dense depth map of the scene. Typically, a surface interpolation or approximation method is used as a means of calculating the dense depth map from its sparse counterpart. A surface-approximation model may be formulated to provide desirable image properties, (such as the lack of additional zero crossings, in the Laplacian of the image irradiances, that are artifacts of the surface approximation model), but often the surface model is based on a priori requirements for the fitted surface, such as smoothness.

The problems encountered in Steps one and two – recovery of the relative orientation of the images and computation of the sparse depth map – are dominated by the problems of image matching. False matches that arise from repetitive scene structures, such as windows of a building, or from image features that are not distinctive (at least, on the basis of local evidence) occur more frequently in the unconstrained matching environment than in the constrained environment. Fortunately, in recovering the relative orientation of the images, we can use redundant information in an effort to reduce the influence of false matches. This is not the case when the sparse depth map is computed. While constrained matching is less susceptible to false matches than is unconstrained matching, there is no redundant information that can be used to identify problems. Furthermore, we have little choice as to which features we may use for sparse depth mapping; if we choose not to use a feature, we cannot recover the relative depth at that scene point.

Selection of suitable features for determining image correspondence is difficult in itself. Correlation techniques embed assumptions that are often violated by the best image features. Area-based correlation techniques usually reflect the premise that image patches are of a scene structure that is all at one distinct depth, whereas edges that arise at object's boundaries are surrounded by surfaces at different scene depths. Edge-based techniques are based on the assumption that an edge found in one image is not "moved" by the change in viewing position of

The research reported herein was supported by the Defense Advanced Research Projects Agency under Contract MDA903-83-C-0027 and by the National Aeronautics and Space Administration under Contract NASA 9-16664. These contracts are monitored by the U.S. Army Engineer Topographic Laboratory and by the Texas A&M Research Foundation for the Lyndon B. Johnson Space Center.

the second image, whereas zero crossings found at boundaries of objects whose gradients are tangential to the line of sight, contradict this assumption. These would seem minor problems, were it not for the accuracy required of the matching process. Typically, the spatial resolution of disparity measurements must be an order of magnitude better than the image's spatial resolution. Matching appears to require distinct features whose properties are incompatible with the assumptions needed to implement the matching process.

The third step, derivation of a dense depth map from a sparse one, is barely adequate. While the stereo pair of images have been used to compute the sparse depth map, they have generally been ignored when the dense surface is being filled in. The dense depth map should, in principle, form the basis for being capable of reproducing the stereo pair of images. The computation of the dense depth map should make explicit use of the stereo irradiance data.

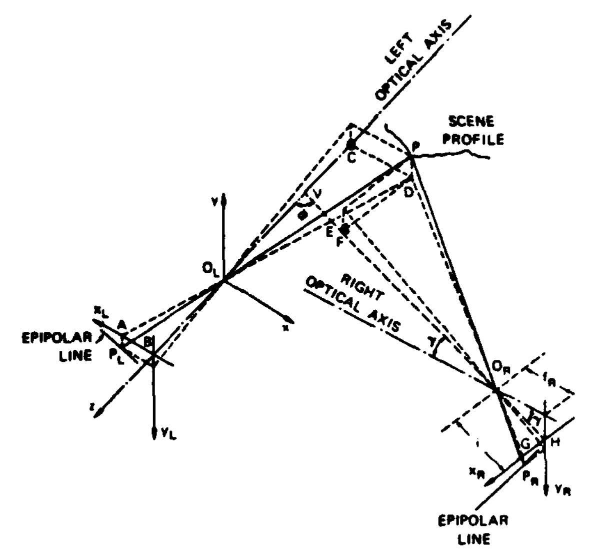
We have previously presented an alternative view of the processes needed to recover the relative orientation of the images. Those ideas are based on symbolic matching of descriptions of linear structures found in the pair of images [1]. In this paper, assuming that we have recovered the relative orientation of the images, we offer a new prescription for Steps two and three - i.e., recovery of a dense relative-depth map of the scene. We use image irradiance profiles as input to an integration routine that returns the corresponding dense relative-depth profile. Our procedure neither matches image points (at least in the conventional sense), nor does it "fill in" data to obtain the dense depth map.

First, we show how we extract "corresponding" irradiance profiles from a stereo pair of images. This is the epipolar mapping that allows stereo reconstruction to b treated as a set of one-dimensional problems. Next, we formulate the one-dimensional integration procedure that returns relative depth. This is the main result presented in this paper. Finally, we show the results we have obtained in applying our technique, and discuss their implications.

It should be noted that, while we phrase this presentation in terms of stereo reconstruction, there is no restriction on the positions of acquisition of the two images; they may equally well be frames from a motion sequence.

2. "Corresponding" Image Irradiance Profiles

The integration procedure takes two image irradiance profiles - one from the left image, one from the right - and computes the corresponding relative-depth profile of the scene. In this section we give a precise definition of image irradiance profile and describe a method for extracting "corresponding" irradiance profiles. These are basically the epipolar mapping considerations, but they provide a means of introducing our notation and establishing the one-dimensional situation analyzed in the next section.



NOTE: A, B, C, D, E, F, G, H, O_L, O_R, and V are coplaner.

Figure 1 Optical Arrangement. A world-coordinate system specified relative to the left imaging system.

We could select any coordinate frame to describe scene depth, provided that we know the position and orientation of the optical systems relative to that frame. Without loss of generality, we will select a particular frame based on the optical arrangement of the left imaging system. Scene depth recovered in this frame may be transformed into any desired frame of reference.

Consider Figure 1. Two optical systems are shown: the left system and the right. We consider a scene depth profile that is the intersection of a plane, an epipolar plane, through the two optical centers, O_L and O_R, and the typical point, P, in the scene. Such a plane generates a one-dimensional matching problem. By rotating this epipolar plane about the axis through the two optical centers, we can build up the two-dimensional scene depth map by recovering the one-dimensional depth profiles.

The coordinate system we adopt is based on the optical arrangement of the left imaging system. The optical axis of the left system defines the z axis. The positive z direction is from world to image, with the optical center of the left system, O_L, being the origin. The x and y coordinate axes lie in a plane parallel to the left image plane. In addition, the x axis lies in the epipolar plane that contains the optical axis of the left imaging system. The y axis is orthogonal to the $x - z$ plane. The positive directions for x and y have been selected to result in a right-handed frame of reference.

The x_L and y_L image-coordinate axes for the left optical system are parallel to the x and y axes, respectively. Their directions, as shown in Figure 1, have been selected so that positive x and y scene coordinates project to positive x_L and y_L image coordinates.

The x_R and y_R image-coordinate axes for the right optical system lie in the right image plane. In addition, the x_R axis lies in the epipolar plane containing the optic axis of the left imaging system. The y_R axis is orthogonal to the x_R axis and is selected so that it passes through the principal point of the right image, the point at which the optical axis of the right imaging system pierces the right image plane. The positive directions of the x_R and y_R axes are selected so that both they and their left-image counterparts, x_L and y_L, would have the same sense if the optical axes of the two imaging system were parallel and both systems viewed the same scene (i.e., they do not face in opposite directions). These directions for x_R and y_R are as shown in Figure 1.

If γ is the angle between the optical axis of the right imaging system and the line O_RV, obtained by rotating the right optical axis about an axis through O_R parallel to the x_R axis, then

$$i = \frac{f_R}{\cos\gamma} \quad , \tag{1}$$

where f_R is the distance from the optical center of the right imaging system to the image plane. Note that the distance i, which enters into the analysis in Section 3, is a function of the optical arrangement and is independent of the scene profile we are considering.

Consider now the epipolar plane that contains the point P. This plane slices the scene, its intersection with the scene being the depth profile we wish to recover. It also slices the two images, the intersections being two "epipolar" lines. Consider for the moment only the left image. The image irradiance in the left image is a function of x_L and y_L, but along the epipolar line under consideration y_L is a function of x_L. Hence, along the epipolar line, the image irradiance is a function of x_L only, that is, the image irradiance profile is $I(x_L)$. Similarly, for the right image along the epipolar line, the image irradiance profile is only a function of x_R, that is, the image irradiance profile is $I(x_R)$. These two irradiance profiles, viewed as functions of the particular coordinates x_L and x_R, are our definition of "corresponding" image irradiance profiles. It may be useful to think of them as image irradiances from epipolar lines that have been projected onto their respective x_L or x_R axis.

Figure 1 illustrates the three-dimensional arrangement of the optical systems. However, if we draw just the two-dimensional arrangement as seen in the epipolar plane that contains the optical axis of the left imaging system, we have the situation in Figure 2. The circumstances depicted in Figure 2 are the same for any "corresponding" image irradiance profiles when these are described as functions of x_L and x_R. Consequently, the following analysis of the situation shown in Figure 2 is independent of the epipolar plane used. Once a depth profile of the scene has been recovered (using the algorithm presented below), this profile can be related to others simply as a function of the angle between the epipolar plane and the optical axis of the left imaging system.

3. Recovery of Relative Depth

The geometrical arrangement presented in Figure 2 allows us to derive expressions relating the world coordinates of the scene to the image coordinates of its projection. The similar triangles ABO_L and CDO_L, along with those of GHO_R and FDO_R, allow us write $\frac{AB}{O_LB} = \frac{CD}{CO_L}$, and hence

$$\frac{x_L}{f_L} = \frac{x}{-z} \quad . \tag{2}$$

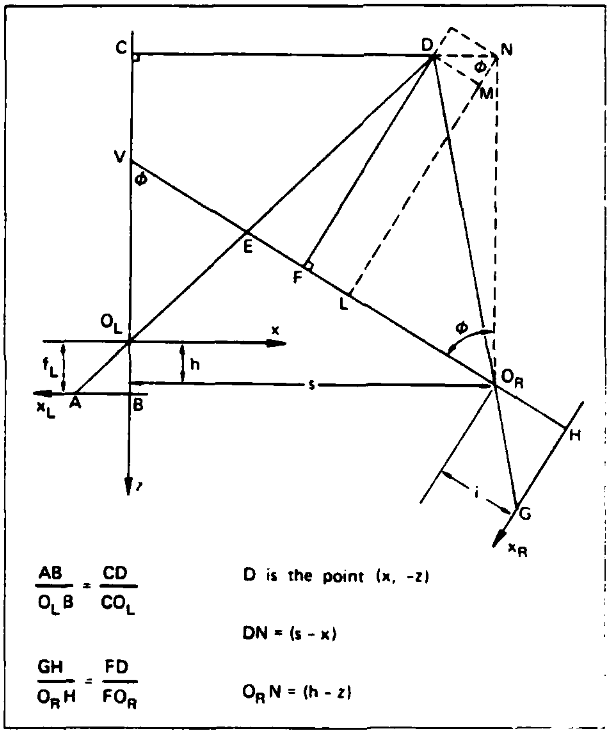


Figure 2 Reduced Model. The two-dimensional arrangement in the epipolar plane that contains the optical axis of the left imaging system. The labels of the vertices correspond to those of Figure 1.

Also $\frac{GH}{O_RH} = \frac{FD}{FO_R}$, but, $\frac{FD}{FO_R} = \frac{LN-MN}{LO_R+MD} = \frac{O_RN\sin\phi - DN\cos\phi}{O_RN\cos\phi + DN\sin\phi}$, $DN = (s-x)$, and $O_RN = (h-z)$, yielding

$$\frac{x_R}{i} = \frac{(h-z)\sin\phi - (s-x)\cos\phi}{(h-z)\cos\phi + (s-x)\sin\phi} \quad . \tag{3}$$

Solving Equations (2) and (3) for x and z, and then using Equation (1) to remove the parameter i, we obtain expressions for the world coordinates of a scene point in terms of image-measurable quantities and the imaging parameters that specify the relative orientation of the two images. The equations are the usual ones obtained from the stereo geometry:

$$x = x_L \frac{(x_R s\cos\gamma - f_R h)\tan\phi + x_R h\cos\gamma + s f_R}{(x_R x_L\cos\gamma + f_R f_L)\tan\phi - x_R f_L\cos\gamma + x_L f_R} \quad . \tag{4}$$

and

$$z = -f_L \frac{(x_R s\cos\gamma - f_R h)\tan\phi + x_R h\cos\gamma + s f_R}{(x_R x_L\cos\gamma + f_R f_L)\tan\phi - x_R f_L\cos\gamma + x_L f_R} \quad . \tag{5}$$

Equations (4) and (5) form part of the algorithm we present. Equations (2) and (3) are used as part of our analysis of the image irradiance information available to us in the two images.

We turn our attention to scene radiance. From a scene point, rays of light proceed to their image projections. What is the relationship between the scene radiance of the rays that project into the left and right images? Let us suppose that the angle between the two rays is small. The bidirectional reflectance function of the scene's surface will vary little, even when it is a complex function of the lighting and viewing geometry. Alternatively, let us suppose that the surface exhibits Lambertian reflectance. The scene radiance is independent of the viewing angle; hence the two ray will have identical scene radiances, irrespective of the size of the angle between them. For the model presented here, we assume that the scene radiance of the two rays emanating from a single scene point is equal. This assumption is a reasonable one when the scene depth is large compared with the separation distance between the two optical systems, or when the surface exhibits approximate Lambertian reflectance. It should be noted that there are no assumptions about albedo (e.g., it is not assumed to be constant across the surface) and, in fact, it is not even necessary to know or calculate the albedo of the surface. Since image irradiance is proportional to scene radiance, we can write, for corresponding image points,

$$I_L(x'_L) = I_R(x'_R) \quad .$$

I_L and I_R are the image irradiance measurements for the left and right images. It should be understood that these measurements at positions x'_L and x'_R are measurements at image points that correspond to a single scene point.

Differentiating the above equation gives

$$\frac{dI_L}{dx}(x'_L) = \frac{dI_R}{dx}(x'_R) \quad ,$$

and hence

$$\frac{dI_L}{dx_L}(x'_L)\,\frac{dx_L}{dx} = \frac{dI_R}{dx_R}(x'_R)\,\frac{dx_R}{dx} \quad .$$

Expressions for $\frac{dx_L}{dx}$ and $\frac{dx_R}{dx}$ are obtained by differentiating Equations (2) and (3).

$$\frac{dx_L}{dx} = -\frac{f_L + x_L\frac{dz}{dx}}{z} \quad , \tag{6}$$

$$\frac{dx_R}{dx} = \frac{x_R \tan\phi\cos\gamma + f_R + (x_R\cos\gamma - f_R\tan\phi)\frac{dz}{dx}}{(h-z)\cos\gamma + (s-x)\tan\phi\cos\gamma} \quad . \tag{7}$$

Substituting these into the previous equation and rearranging terms, we obtain an expression for $\frac{dz}{dx}$, namely

$$\frac{dz}{dx} = -\frac{\left(\frac{dI_L}{dx_L} f_L \cos\gamma(h-z+(s-x)\tan\phi) + \frac{dI_R}{dx_R} z(x_R\tan\phi\cos\gamma + f_R)\right)}{\left(\frac{dI_L}{dx_L} x_L \cos\gamma(h-z+(s-x)\tan\phi) + \frac{dI_R}{dx_R} z(x_R\cos\gamma - f_R\tan\phi)\right)} \quad . \tag{8}$$

Note that, for clarity of expression, we have dropped the notation (x'_L) and (x'_R) that shows the value of the independent variable at which the image irradiance gradients are to be evaluated. All terms that involve the image irradiance are understood to be evaluated at corresponding image points.

We are now ready to outline an algorithm to recover scene depth:

1. Suppose we have a pair of corresponding image points, x_L and x_R. We use Equations (4) and (5) to calculate x and z for the scene point.
2. Equation (8) is used to calculate $\frac{dz}{dx}$ for this scene point.
3. Equations (6) and (7) are used to calculate a dx_R for a choosen dx_L.
4. The pair of points $x_L + dx_L$ and $x_R + dx_R$ are corresponding image points; Steps 1 to 3 may be repeated.

This, then, is an integration procedure that, given an initial pair of corresponding image points, proceeds along the two image irradiance profiles, maintaining correspondence. As in other numerical integration procedures, we can adjust the step size dx_L so that the scene's profile gradient, $\frac{dz}{dx}$, varies slowly between successive steps. In the following section we shall discuss the application of this algorithm to scene profiles that have discontinuities.

An obvious difficulty with the algorithm, as outlined, occurs when both $\frac{dI_L}{dx_L}$ and $\frac{dI_R}{dx_R}$ are zero; $\frac{dz}{dx}$ is indeterminant. A solution is still possible if the second derivatives of image irradiance are not zero as well. Differentiating $I_L = I_R$ twice gives us

$$\frac{d^2I_L}{dx_L{}^2}\left(\frac{dx_L}{dx}\right)^2 + \frac{dI_L}{dx_L}\frac{d^2x_L}{dx^2} = \frac{d^2I_R}{dx_R{}^2}\left(\frac{dx_R}{dx}\right)^2 + \frac{dI_R}{dx_R}\frac{d^2x_R}{dx^2} \quad ,$$

which reduces to

$$\sqrt{\frac{d^2I_L}{dx_L{}^2}}\,\frac{dx_L}{dx} = \sqrt{\frac{d^2I_R}{dx_R{}^2}}\,\frac{dx_R}{dx} \quad ,$$

when $\frac{dI_L}{dx_L}$ and $\frac{dI_R}{dx_R}$ are zero. Hence

$$\frac{dz}{dx} = -\frac{\left(\sqrt{\frac{d^2I_L}{dx_L{}^2}} f_L \cos\gamma(h-z+(s-x)\tan\phi) + \sqrt{\frac{d^2I_R}{dx_R{}^2}} z(x_R\tan\phi\cos\gamma + f_R)\right)}{\left(\sqrt{\frac{d^2I_L}{dx_L{}^2}} x_L \cos\gamma(h-z+(s-x)\tan\phi) + \sqrt{\frac{d^2I_R}{dx_R{}^2}} z(x_R\cos\gamma - f_R\tan\phi)\right)} \quad . \tag{9}$$

When $\frac{dI_L}{dx_L}$ and $\frac{dI_R}{dx_R}$ are both zero, we adjust Step 2 of the algorithm to use Equation (9) rather than Equation (8). This allows integration through the peaks and troughs of image irradiance.

It should be noted that scene depth profiles of planar objects have zero image irradiance gradients and zero second derivatives. These situations must be detected and treated separately, for there is no information available, except at the object's boundaries, from which to assess orientation.

The integration routine uses the information available in the geometric distortion of perspective projection. It does not use the reflectance characteristics of the scene, nor does it need to know them. The method is based on the assumption that the scene radiance of two rays emanating from a single scene point (and entering the two optical systems) is equal. Spatial variations in albedo and lighting are inconsequential for this procedure.

4. Experimental Results and Discussion

The presented algorithm requires as input spatially-continuous image irradiance profiles. To apply it to digital

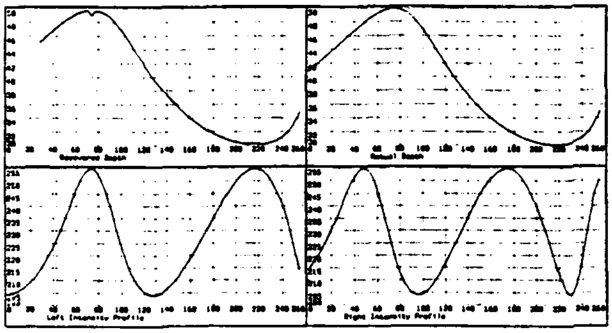


Figure 3 Depth Recovery - Ideal Case. Upper left shows the recovered depth from the two irradiance profiles shown in the lower half. For comparison, the actual depth is shown in upper right.

images we must first construct spatially-continuous profiles from their sampled counterparts. We use simple modelling techniques, such as linear interpolation, to do this.

The result of applying the above algorithm to two synthetic, corresponding Lambertian image irradiance profiles is shown in Figure 3. The actual depth profile corresponding to the irradiance profiles is shown in the upper right portion of Figure 3. For this example, initial starting positions for the integration were selected near the center of each profile. These initial positions were corresponding points, with no error in the determination of their location. The integration process was applied in both directions from the initial point. The recovered depth is shown in the upper left part of Figure 3.

A second example is shown in Figure 4. The image irradiance profiles were obtained by "painting" the previous surface with "pigment" of continuously varying albedo. In addition, three strips of different albedos were painted on the surface. The effect can be seen by examining the image irradiance profiles shown in the bottom half of Figure 4. The processes we applied to recover depth were twofold. First, we used a simple smoothing routine, based on moving average, to produce intermediate profiles. This rounded the step edges associated with the albedo strips. Next the integration procedure was applied. The result is shown in the upper left part of Figure 4.

You will notice small errors near the peaks and troughs of irradiance, where second derivative information is being used. Furthermore, there are small errors associated with albedo edges. What is happening here is that the tracking mechanism that maintains point correspondence as it moves along the profiles is getting out of step. The process is "self-correcting", however, a feature that we will exploit in the next example. Note that the continuously variable albedo change across the profiles has no influence on the resulting recovered depth.

What would be the effect if the initial matched points were in error? We repeat the above procedure but select initial starting points that are mismatched by two pixels (the horizontal units in Figures 3, 4 and 5). The left half of Figure 5 demonstrates the result achieved. The effect of the starting point error shows up as depth error at positions 120 to 130 on the horizontal axis. Note the swift correcting action, which suggests that the initial points are not critical for the recovery of depth. Clearly, this algorithm has a very special feature whose implication for stereo processing is far-reaching: approximate matches

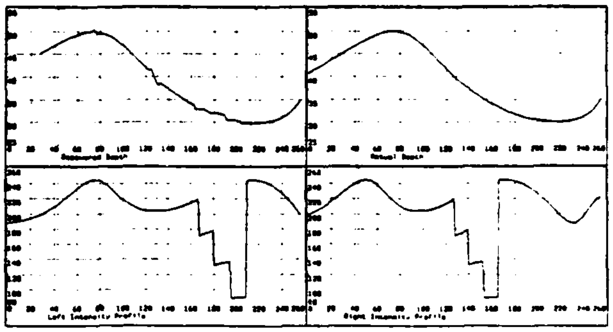


Figure 4 Depth Recovery - Painted Surface.

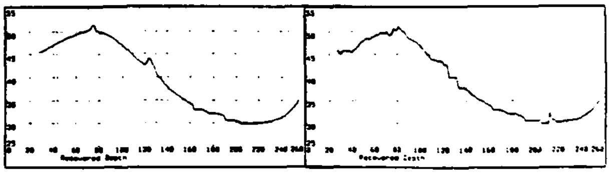


Figure 5 Depth Recovery - Mismatched Initial Points, and Noise Concerns.

are all that is necessary for the recovery of scene depth.

The above examples have been based on synthetic images. We now turn our attention to real scenes that are full of discontinuities in the depth profile, and to real images that are not free of noise.

In the synthetic scene profile used in the previous examples, we have used continuous-depth profiles. For real scenes this is unrealistic. At an object's boundaries, discontinuities in depth are likely. Because the presented algorithm cannot integrate across these discontinuities, we need to be able to identify them. Let us suppose that we use zero crossings of the Laplacian of image irradiance as places at which depth discontinuities may occur. We will apply our integration procedure, tracking along the image irradiance profiles until we come to a zero-crossing in one of the image irradiance profiles.

If continuation implies that the scene depth gradient, $\frac{dz}{dx}$, varies slowly, then we continue. A sudden change in gradient signals a depth discontinuity and the integration procedure is terminated. Note that the integration routine itself signals depth discontinuity if $\frac{dz}{dx}$ exhibits rapid change for arbitrarily small step sizes. This procedure also handles occlusion problems in which one view (hence its image irradiance profile) "sees" around an object that is occluded from the other view. Again we stop at the first zero crossing encountered in either of the image irradiance profiles, or when $\frac{dz}{dx}$ changes too rapidly. It should be noted that the above procedure does not require that the zero crossing from both image irradiance profiles be matched; rather, it simply requires their detection.

Of course, there is a price that must be paid: we now need to be able to detect initial starting points for the integration procedure between adjacent zero crossings. The peaks and troughs of image irradiance would seem appropriate, being invariant through most realistic image irradiance transformations that may occur during image acquisition. Furthermore, as these peaks and troughs of the two irradiance profiles match, and since the value of irradiance should be identical at matched points, the opportunity exists for correcting the image irradiances for linear transformations in contrast. This allows for local contrast correction. A suggested procedure is to (1) detect the peaks and troughs in image irradiance, and also the zero crossings of the Laplacian of image irradiance; (2) match the peaks and troughs across the two images to provide initial points for integration;[1] (3) correct the image irradiance profiles for each profile section between peaks and troughs for a linear transformation in contrast; (4) then apply the integration procedure, terminating at rapid changes in $\frac{dz}{dx}$ or at zero crossings, if necessary. We are currently giving our attention to these matters.

A serious deficiency of the present algorithm is its sensitivity to noise – a disadvantage inherent in any procedure that makes use of image irradiance gradients. This sensitivity can be easily demonstrated with quantization noise alone. If the image irradiances shown in Figure 4 are quantized to 256 different levels, the results of applying the algorithm can be seen in the right half of Figure 5. This result should be compared with the one shown in the upper left of Figure 4. Noise is an undeniable problem. We have difficulty in recovering reliable depth estimates if the signal-to-noise ratio is less than a few hundred. This sensitivity is particularly apparent when the image irradiance gradient is small. Smoothing of the image irradiance profiles is at best inadequate. We are actively addressing this problem in our current research. A solution is necessary if the presented algorithm is to become a viable technique for recovering scene depth from stereo pairs of real images that cannot be preprocessed to remove noise.

5. Summary

We have presented a new approach to reconstruction of scene depth from a stereo pair of images. The technique does not depend upon matching of image features, at least not in the usual sense, and the necessary matching does not require great spatial accuracy. Furthermore, the features to be matched are more compatible than their traditional counterparts with the assumptions implicit in correlation techniques.

The results point to a technique that is capable of handling changes in both albedo and illumination. Furthermore, the technique directly yields a dense depth map of the scene.

We are exploring several related outstanding issues. Among these are the exploitation of depth discontinuities and the problem of reducing sensitivity to image noise.

References

1. Smith, G.B. and Wolf, H.C., Image-to-Image Correspondence: Linear-Structure Matching, *Proceedings of Second Annual NASA Symposium on Mathematical Pattern Recognition and Image Analysis*, Houston, Texas, 1984, pp 467-487.

[1] We do not underestimate the difficulty of this step, but the basic assumptions implicit in correlation techniques are likely to be met near peaks and troughs. Some mismatch error can be tolerated and as we can integrate through peaks and troughs of image irradiance, we have only to detect and match the "obvious" ones.

Appendix C

One-Eyed Stereo: A General Approach to Modeling 3-D Scene Geometry

By: Thomas M. Strat and Martin A. Fischler

One-Eyed Stereo: A General Approach to Modeling 3-D Scene Geometry [1]

Thomas M. Strat and Martin A. Fischler
Artificial Intelligence Center
SRI International
Menlo Park, California

Abstract

A single two-dimensional image is an ambiguous representation of the three-dimensional world—many different scenes could have produced the same image—yet the human visual system is extremely successful at recovering a qualitatively correct depth model from this type of representation. Workers *in the field of computational vision* have devised a number of distinct schemes that attempt to emulate this human capability; these schemes are collectively known as "shape from ...." methods (*e.g.*, shape from shading, shape from texture, or shape from contour). In this paper we contend that the distinct assumptions made in each of these schemes must be tantamount to providing a second (virtual) image of the original scene, and that any one of these approaches can be translated into a conventional stereo formalism. In particular, we show that it is frequently possible to structure the problem as one of recovering depth from a stereo pair consisting of the supplied perspective image (the *original* image) and an hypothesized orthographic image (the *virtual* image). We present a new algorithm of the form required to accomplish this type of stereo reconstruction task.

[1] The work reported herein was supported by the Defense Advanced Research Projects Agency under Contract MDA903-83-C-0027.

1 Introduction

The recovery of 3-D scene geometry from one or more images, which we will call the scene-modeling problem (SMP), has solutions that appear to follow one of three distinct paradigms: stereo; optic flow; and shape from shading, texture, and contour.

In the stereo paradigm, we match corresponding world/scene points in two images and, given the relative geometry of the two cameras (eyes) that acquired the images, we can use simple trigonometry to determine the depths of the matched points [1].

In the optic-flow paradigm, we use two or more images to compute the image velocity of corresponding scene points. If the camera's motion and imaging parameters are known, we can again use simple trigonometry to convert velocity measurements in the image to depths in the scene [21].

In the shape from shading, texture, and contour (SSTC) paradigm, we must either know, or make some assumptions about the nature of the scene, the illumination, and the imaging geometry. Brady's 1981 volume on computer vision [2] contains an excellent collection of papers, many of which address the problem of how to recover depth from the shading, texture, and contour information visible in a single image. Two distinct computational approaches have been employed in the SSTC paradigm: (1) integration of partial differential equations describing the relation of shading in an image to surface geometry in a scene, and (2) back-projection of planar image facets to undo the distortion in an image attribute (*e.g.*, edge orientation) induced by the imaging process on an assumed scene property (*e.g.*, uniform distribution of edge orientations).

Our purpose in this paper is to provide a unifying framework for the scene modeling problem, and to present a new computational approach to recovering scene geometry from the shading, texture, and contour information in a single image. Our contribution is based on the following observation: regardless of the assumptions employed in the SSTC paradigm, if a 3-D scene model has been derived successfully, it will generally be possible to establish a large number of correspondences between image and scene (model) points. From these correspondences we can compute a collineation matrix [11], and then extract the imaging geometry from it [4] [19]. We can now construct a second image of the scene as viewed by the camera from some arbitrary location in space. It is thus obvious that any technique that is competent to solve the SMP must either be provided with at least two images or make assumptions that are equivalent to providing a second image. We can unify the various approaches to the SMP by converting their respective assumptions and auxiliary information into the implied second image and employing the stereo paradigm to recover depth. In the case of the SSTC paradigm, our approach amounts to "one-eyed stereo."

2 Shape from One-Eyed Stereo

Most people viewing Figure 1 get a strong impression of depth. We can recover an equivalent depth model by assuming that we are viewing a projection of a uniform grid and employing the computational procedure to be described. In the remainder of this paper we will show how some simple modifications and variations of the uniform grid, as the implied second image, allow us to recover depth from shading, texture, and contour.

The one-eyed stereo paradigm can be described as a five-step process, as outlined in the paragraphs below. Some scenes with special surface markings or image-formation processes must be analyzed by variants of the algorithm described, but the general approach remains the same.

2.1 Partition the Image

As with all approaches to the SMP, the image must be segmented into regions prior to the application of a particular algorithm. Before the one-eyed stereo computation can be employed, the segmentation process must delineate regions that are individually in conformance with a single model of image formation. The computation can then be carried out independently in each region, and the results fitted together.

2.2 Select a Model

For each region identified by the partitioning process, we must determine the underlying model of image formation that explains that portion of the image. Surface reflectance functions and texture patterns are examples of such models. Partitioning of the image and selection of the appropriate models are difficult tasks that are not addressed in this paper. Witkin and Kass [23] are exploring a new class of techniques that promises some eventual answers to these questions. Generally, it will be impossible to recover depth whenever a single model cannot be associated with a region. Similarly, inaccurate or incorrect results can be expected if the partitioning or modeling is performed incorrectly.

2.3 Generate the Virtual Image

The key to one-eyed stereo is using the model of image formation to fabricate a second (virtual) image of the scene. The idea is that the model often allows one to construct an image that is independent of the actual shape of the imaged surface. This allows the virtual image to be determined solely from knowledge of the model without making use of the original image. For example, the markings on the surface of Figure 2(a) could have arisen from projection of a uniform grid upon the surface. For all images that fit this model, we can use a uniform grid as the virtual image. As a rule, the orientation, position, and scale of this grid will be unknown; however, we will show how this information can be recovered from the original image. Other models give rise to other forms of virtual images.

2.4 Determine Correspondences

Before applying stereo techniques to determine depths, we must first establish correspondences between points in the real image and the virtual image. When dealing with textures, the process is typified by counting texels in each image from a chosen starting point. With shaded images, the general approach is to integrate intensities. Several variants of the method for establishing correspondences are described in the next section. The difficulty of the procedure, it should be noted, will depend on the nature of the model.

2.5 Compute Depths Using Stereo

With two images and a number of point-to-point correspondences in hand, the techniques of binocular stereo are immediately applicable. At this point, the problem has been reduced to computing the relative camera models between the two images and using that information to compute depths by triangulation. The fact that the virtual image will normally be an orthographic projection required reformulation of existing algorithms for performing this computation. The appendix describes a new algorithm that computes the relative camera model and reconstructs the 3-D scene from eight point correspondences between a perspective and an orthographic image.

The problem of recovering scene and imaging geometry from two or more images has been addressed by workers not only in binocular stereo, but also in monocular perception of motion in which the two projections are separated in time as well as space. Various approaches have been employed to derive equations for the 3-D coordinates and motion parameters; these equations are generally solved by iterative techniques [5] [8] [13] [14]. Ullman [21] presents a solution for recovering 3-D shape from three orthographic projections with established correspondences among at least four points. His "polar equation" allows computation of shape when the motion of the scene is restricted to rotation about the vertical axis with arbitrary translation. Nagel and Neumann [10] have devised a compact system of three nonlinear equations for the unrestricted problem when five point correspondences between the two perspective images are known. More recently, Huang [20] and Longuet-Higgins [9] have independently derived methods requiring only that a set of eight simultaneous linear equations be solved when eight point correspondences between two perspective images are known. In our formulation we are faced with a stereo problem involving a perspective and an orthographic image; while the aforementioned references are indeed germane, none provides a solution to this particular problem.

The derivation described in the appendix was inspired by the formulation of Longuet-Higgins for perspective images. When either image nears orthography, Longuet-Higgins's method becomes unstable; it is undefined if either image is truly orthographic. Moreover, his approach requires knowledge of the focal length and principal point in each image while our method was derived specifically for one orthographic and one perspective image whose internal imaging parameters may not be fully known.

3 Variations on the Theme

In this section we illustrate how our approach is used with several models of texture, shading, and contour. Where these models do not match given scene characteristics, they may require additional modification. However, a qualitatively correct answer might still be obtainable by applying one of the specific models we discuss below to a situation that appears to be inappropriate, or to an image in which the validity of the assumptions cannot be established.

3.1 Shape from Texture

Surface shapes are often communicated to humans graphically by drawings like Figure 2(a). Such illustrations can also be interpreted by one-eyed stereo. In this case, there is no need to partition the image; the underlying model of the entire scene consists of the intersections of lines distributed in the form of a square grid. When viewed directly from above at an infinite distance, the surface would appear as shown in the virtual image of Figure 2(b) regardless of the shape of the surface. This virtual image can be construed as an orthographic projection of the object surface from a particular, but unknown, viewing direction. Correspondences between the original and virtual images are easily established if there are no occlusions in the original image. Select any intersection in the original image to be the reference point and pair it with any intersection in the virtual image. A second corresponding pair can be found by moving to an adjacent intersection in both images. Additional pairs are found in the same manner, being careful to correlate the motions in each image consistently in both directions. When occlusions are present, it may still be possible to obtain correspondences for all visible junctions by following a nonoccluded path around the occlusion (such as the hill in the foreground of Figure 2(a)). If no such path can be found, the shape of each isolated region can still be computed, but there will be no way to relate the distances without further information. Other techniques used to represent images of 3-D shapes graphically may require other virtual images. Figure 3(a), for example, would imply a virtual image as shown in Figure 3(b). Methods for recognizing which model to apply are needed, but are not discussed here.

Once correspondences have been determined, we can use the algorithm given in the appendix to recover depth. We have presumably one perspective image and one orthographic image whose scale and origin are still unknown. The depths to be recovered will be scaled according to the scale chosen for the virtual image[2]. The choice of origin for the orthographic image is arbitrary, and will lead to the same solution regardless of the point chosen. The appendix shows how to compute both the orientation and the displacement of the orthographic coordinate system, relative to the perspective imaging system. 3-D coordinates of each matched point are then easily computed by means of back-projection. A unique solution will be obtained whenever the piercing point or focal length of the perspective image is known. A minimum of eight pairs of matched points is required to obtain a solution; depths can be computed for all matched points.

There exists a growing literature on methods to recover shape from natural textures [7][12][18][22]. We will now show how the constraints imposed by one type of natural texture can be exploited to obtain similar results by using one-eyed stereo.

Consider the pattern of streets in Figure 4. If this city were viewed from an airplane directly overhead at high altitude, the streets would form a regular grid not unlike the one used as the virtual image in Figure 2. There are many other scene attributes that satisfy this same model. The houses in Figure 5 would appear to be distributed in a uniform grid if viewed from directly overhead. In an apple orchard growing on a hillside, the trees would be planted in rows that are evenly spaced when measured horizontally; the vineyard in Figure 6 exhibits this property.

Ignoring the nontrivial tasks of partitioning these images into isotextural regions, verifying that they satisfy the model, and identifying individual texels, it can be seen how these images can be interpreted with the same

[2] Recall that the original image does not contain the information necessary to recover the absolute size of the scene.

techniques as were described in the previous section. The virtual image in each case will be a rectangular grid that can be considered as an orthographic view from an unknown orientation. Correspondences can be established by counting street intersections, rooftops, or grape vines. As before, one can solve for the relative camera model and compute depths of matched points. Obviously, for the situations discussed here, we must be satisfied with a qualitatively correct interpretation—not only because of the difficulty of locating individual texels reliably and accurately, but also in view of the numerical instabilities arising from the underlying nonlinear transformation.

3.2 Shape from Shading

For our purposes, surface shading can be considered the limiting case of a locally uniform texture distribution (as the texels approach infinitesimal dimensions). To compute correspondences, we need to integrate image intensities appropriately in place of counting lines, since the image intensities can be seen to be related to the density of lines projected on the surface. The feasibility of this procedure depends on the reflectance function of the surface.

What types of material possess the special property that allows their images to be treated like the limiting case of the projected textures of the previous section? The integral of intensity in an image region has to be proportional to the number of texels that would be projected in that region. If the angles i and e are defined as depicted in Figure 7, it can be seen that the number of texels projected onto a surface patch will be proportional to $\cos i$, the cosine of the incident angle. At the same time, the surface patch (as seen from the viewpoint) will be foreshortened by $\cos e$, the cosine of the emittance angle. Thus, the integral of reflected light intensity over a region will be proportional to the flux of the light striking the surface if the intensity of the reflected light at any point is proportional to $\cos i/\cos e$. Horn [6] has pointed out that, when viewed from great distances, the material in the maria of the moon and other rocky, dusty objects exhibit a reflectance function that allows recovery of the ratio $\cos i/\cos e$ from the imaged intensities. This surface property has made possible unusually simple algorithms for computing shape-from-shading, so it is not surprising that it submits easily to one-eyed stereo as well.

To interpret this type of shading, we can construct a virtual image whose direction of view is the lighting direction (*i.e.*, taken from a "virtual camera" located at the light source). When the original shaded image is orthographic, we consider a family of parallel lines in which each line lies in a plane that includes both the light source and the (distant) viewpoint. When viewed from the light source, the image of the surface corresponding to these lines will also be a set of parallel lines regardless of the shape of the surface. These parallel lines constitute the virtual image. We will use the image intensities to refine these line-to-line correspondences to point-to-point correspondences. Figure 8 shows the geometry for an individual line in the family. A little trigonometry shows that

$$\Delta s' = \frac{\cos i}{\cos e} \Delta s \qquad , \tag{1}$$

where Δs is a distance along the line in the real image and $\Delta s'$ is the corresponding distance along the corresponding line in the virtual image. Integrating this equation produces the following expression, which defines the point correspondences in the two images along the given line.

$$s' = s'_o + \int_0^s \frac{\cos i}{\cos e} ds \tag{2}$$

To use this equation we must first compute $\frac{\cos i}{\cos e}$ from the intensity value at each point along the line. This will, of course, be possible only when the reflectance function is constant for constant $\frac{\cos i}{\cos e}$. Next we choose a starting point in the shaded image and begin integrating intensities according to Equation (2). For any value of s, the corresponding virtual image point is along a straight line at a distance s' from the virtual reference point. With these point-to-point correspondences in hand, it is a simple matter of triangulation to find the 3-D coordinates of the surface points, given that we know the direction to the light source. We can explore the remainder of the surface by repeating the process for each of the successive parallel lines in the image. Adjacent profiles still remain unrelated to each other, since their individual scale factors have not yet been determined. Knowledge of the actual depth of one point along each profile provides the necessary additional information to complete the reconstruction. It is important to note that our assumptions and initial conditions are those used by Horn; the fact that he was able to obtain a solution under these conditions assured the existence of a suitable virtual image for the one-eyed stereo paradigm.

For shaded perspective images, we must integrate along a family of straight lines that radiate from the point in the image that corresponds to the location of the light source. This ensures that the image line will be in a plane containing both the viewer and the light source, and that the virtual image of each line will also be a straight line. The integration becomes a bit more complex than shown in Equation 2 because the nonlinear effects of perspective imaging must be accommodated. Nevertheless, it remains possible to establish point-to-point correspondences between images and to reconstruct the surface along each line.

3.3 Shape from Contour

It is sometimes possible to extract a line drawing, such as the one shown in Figure 9, from scene textures. Parallel streets like those encountered in Figure 4 give rise to a virtual image consisting of parallel lines when the cross streets cannot be located; terraced hills also produce a virtual image of parallel lines. Correspondences between real and virtual image lines can be found by counting adjacent lines from an arbitrary starting point. This matches a virtual image line with each point in the real image. Point-to-line correspondences are not sufficient to enable the stereo computation of the appendix to be used for reconstruction of the surface. Knowledge of the relative orientation between the two images (equivalent to knowing the orientation of the camera that produced the real image relative to the parallel lines in the scene) provides an adequate constraint; the surface can then be reconstructed uniquely through back-projection. Without knowledge of the relative orientation of the virtual image, heuristics must be employed that relate points on adjacent contours so that a regular grid can be used as the virtual image. The human visual system is normally able to interpret images like Figure 9 unambiguously although just what assumptions are being made remains unclear. Further study of this phenomenon may make it possible to extract models that are especially suited to the employment of one-eyed stereo on this type of image without requiring prior knowledge of the virtual orientation.

3.4 Distorted Textures and Unfriendly Shading

We have already noted that image shading can be viewed as a limiting (and, for our purposes, a degenerate) result of closely spaced texture elements. To recover depth from shading, we must use integration instead of the process of counting the texture elements that define the locations of the "grid lines" of our virtual image. The integration process depends on the existence of a "friendly" reflectance function and an imaging geometry that allows us to convert distance along a line in the actual image to a corresponding distance along a line in the virtual image.

The recovery of lunar topography from a single shaded image [6], as discussed in Section 3.2, is one of the few instances in which "shape from shading" is known to be possible without a significant amount of additional knowledge about the scene. Nevertheless, even here we are required to know the actual reflectance function, the location of the [point] source of illumination, and the depths along a curve on the object surface, and be dealing with a portion of the surface that has constant albedo. Furthermore, the reflectance function has to have just the property we require to replace direct counting, *i.e.*, the reflectance function has to compensate exactly for the "foreshortening" of distance due to viewing points on the object surface from any angle. Most of the commonly encountered reflectance functions, such as Lambertian reflectance, do not possess this friendly property, and it is not clear to what extent it is possible to recover depth from shading in such cases (*e.g.*, see Pentland [12] and Smith [15]). Additional assumptions will probably be necessary and the qualitative nature of the recovery will be more pronounced. Just as in the case in which a complex function can be evaluated by making a local linear approximation and iterating the resulting solution, so it may be possible to deal with unfriendly, or even unknown, reflectance functions by assuming that they are friendly in the vicinity of some point, solving directly for local shape by using the algorithm applicable to the friendly case, and then extending the solution to adjacent regions. We are currently investigating this approach.

The uniform rectangular grid and the polar grid that we used as virtual images to illustrate our approach to one-eyed stereo are effective in a large number of cases because there are processes operating in the real world that produce corresponding textures (*i.e.*, gridlike textures that appear to be orthographically projected onto the surfaces of the scene). However, there are also textures that produce similar-appearing images, but are due to different underlying processes. For example, a uniform gridlike texture might have been created on a flat piece of terrain that is subsequently subjected to geologic deformation—in this case the virtual image (or the recovery algorithm) needed to recover depth must be different from the projective case. We have already indicated the problem of choosing the appropriate model for the virtual image and, as noted above, image appearance alone is probably insufficient for making this determination—some semantic knowledge about the scene is undoubtedly essential. Figure 10 shows an example in which two completely different, yet equally believable, interpretations of scene structure result, depending on whether we use the rectangular grid model or the polar grid model.

4 Experimental Results

The stereo reconstruction algorithm described in the appendix has been programmed and successfully tested on both real and synthetic imagery. Given a sparse set of image points and their correspondence in a virtual image, a qualitative description of the imaged surface can be obtained.

Synthetic images were created from surfaces painted with computer-generated graphic textures. Figure 11(a) shows a synthetic image constructed from a section of a digital terrain model (DTM). The intersections of every twentieth grid line constitute the set of 36 image points made available to the one-eyed stereo algorithm. Their correspondences were determined by selecting an arbitrary origin and counting grid lines to obtain virtual image coordinates. When these pairs are processed by the algorithm in the appendix, a set of 3-D coordinates is obtained in either the viewer-centered coordinate space, or the virtual image coordinate space (which, if correct, is aligned with the original DTM). Figure 11(b) was produced from the resulting 3-D coordinates expressed in the virtual image space by using Smith's surface interpolation algorithm [16] to fit a surface to these points. This yields a dense set of 3-D coordinates that can then be displayed from any viewpoint. The viewpoint that was computed by one-eyed stereo was used to render the surface as shown in Figure 11(b). Its similarity to the original rendering (Fig. 11(a)) confirms the successful reconstruction of the scene.

The same procedure was followed when we worked with real photographs. The intersections of 31 street intersections were extracted manually from the photograph of San Francisco shown in Figure 4. Those that were occluded or indistinct were disregarded. Virtual image coordinates were obtained by counting city blocks from the lower-left intersection. The one-eyed stereo algorithm was then used to acquire 3-D coordinates of the corresponding image points in both viewer-centered and grid-centered coordinate systems. A continuous surface was fitted to both representations of these points. The location and orientation of the camera relative to the grid were also computed. Figure 12(a) shows the reconstructed surface as an orthographic view from the direction computed to be true vertical. The numbers superimposed are the computed locations of the original 31 points. Figure 12(b) shows the surface from the derived location of the viewpoint of the original photo. While several of the original points were badly mislocated, the general shape of the landform is apparent.

There are several reasons the algorithm can provide only a qualitative shape description. First, the problem itself can be somewhat sensitive to slight perturbations in the estimates of the piercing point or focal length. This appears to be inherent to the problem of recovering shape from a single image. How humans can determine shape monocularly without apparent knowledge of the piercing point or semantic content of the scene remains unresolved. The second factor precluding precise, quantitative description of shape is the practical difficulty of acquiring large numbers of corresponding points. While the algorithm can proceed with as few as eight points, the location of the object will be identified at those eight points only. If a more complete model is sought, additional points will be required to constrain the subsequent surface interpolation.

The task remains to evaluate the effectiveness of the iterative technique, described in Section 3.4, for recovering (1) shape from shading in the case of scenes possessing "unfriendly" reflectance functions, and (2) shape from nonprojective and distorted textures. Our experience with the process indicates that the key to surmounting these problems lies in the ability to establish valid correspondences with the virtual image. With these in hand, reconstruction of the surface can proceed as outlined in the foregoing discussion.

5 Conclusion

In this paper we have shown that, in principle, it is possible to employ the stereo paradigm in place of various approaches proposed for modeling 3-D scene geometry—including the case in which only one image is available. We have further shown that, for the case of a single image, the approach could be implemented by

(1) Setting up correspondences between portions of the image and some variants of a uniform grid, and;

(2) Treating each image region and its grid counterpart as a stereo pair, and employing a stereo technique to recover depth. (We present a new algorithm that makes it possible to accomplish this step.)

Devising automatic procedures to partition the image, select the appropriate form of the virtual image, and establish the correspondences are all difficult tasks that were not addressed in this paper. Nevertheless, we have unified a number of apparently distinct approaches, that individually, also have to contend with these same pervasive problems (*i.e.*, partitioning, model selection, and matching).

References

[1] Barnard, S. T., and Fischler, M. A., "Computational Stereo," Computing Surveys, Vol. 14, No. 4, December 1982.

[2] Brady. M., ed., *Artificial Intelligence* (Special Volume on Computer Vision), Volume 17, Nos. 1–3, August 1981.

[3] Cameron, R., *Above San Francisco*, Cameron and Company, San Francisco, 1976.

[4] Ganapathy, S., "Decomposition of Transformation Matrices for Robot Vision," International Conference On Robotics, (IEEE Computer Society), Atlanta, Georgia, March 13-15, 1984, pp. 130–139.

[5] Gennery, D. B. "Stereo Camera Calibration," Proceedings of the IU Workshop, November 1979, pp. 101–107.

[6] Horn, B. K. P., "Image Intensity Understanding," MIT Artificial Intelligence Memo 335, August 1975.

[7] Kender, J. R., "Shape from Texture," Ph.D. thesis, Carnegie–Mellon University, CMU-CS-81-102, November 1980.

[8] Lawton, D. T., "Constraint-Based Inference from Image Motion," Proc. AAAI-80, pp. 31-34.

[9] Longuet-Higgins, H. C., "A Computer Algorithm for Reconstructing a Scene from Two Projections," Nature, Vol. 293, September 1981, pp. 133–135.

[10] Nagel, H., and Neumann, B., "On 3-D Reconstruction from Two Perspective Views," Proc. IEEE 1981.

[11] Nitzan, D., Bolles, R.C., it et. al., "Machine Intelligence Research Applied to Industrial Automation," 12th Report SRI Project 2996, January 1983.

[12] Pentland, A. P., "Shading into Texture" Proceedings AAAI-84, August 1984, pp. 269–273.

[13] Prazdny, K., "Motion and Structure from Optical Flow," Proc. IJCAI-79, pp. 704-704.

[14] Roach, J. W., and Aggarwal, J. K., "Determining the Movement of Objects from a Sequence of Images," IEEE Trans. on Pattern Analysis and Machine Intelligence, Vol. PAMI-2, No. 6, November 1980, pp. 554–562.

[15] Smith, G. B., "The Relationship between Image Irradiance and Surface Orientation," Proc. IEEE CVPR-83.

[16] Smith, G. B., "A Fast Surface Interpolation Technique," Proceedings: DARPA Image Understanding Workshop, October 1984, pp. 211–215.

[17] Stevens, K. A., "The Line of Curvature Constraint and the Interpretation of 3-D Shape from Parallel Surface Contours," AAAI-83, pp. 1057–1061.

[18] Stevens, K. A., "The Visual Interpretation of Surface Contours," Artificial Intelligence Journal Vol. 17, No. 1, August 1981, pp. 47–73.

[19] Strat, T. M., "Recovering the Camera Parameters from a Transformation Matrix," Proceedings: DARPA Image Understanding Workshop, October 1984, pp. 264–271.

[20] Tsai, R.Y. and Huang, T.S., "Uniqueness and Estimation of Three-Dimensional Motion Parameters of Rigid Objects with Curved Surfaces," IEEE Trans. on Pattern Analysis and Machine Intelligence, vol. PAMI-6, No. 1, Jan 1984, pp. 13-27.

[21] Ullman, S., *The Interpretation of Visual Motion*, The MIT Press, Cambridge, Mass., 1979.

[22] Witkin, A. P., "Recovering Surface Shape and Orientation from Texture," Artificial Intelligence Journal Vol. 17, No. 1, August 1981, pp. 17–45.

[23] Witkin, A., and Kass, M., "Analyzing Oriented Patterns," Proceedings IJCAI-85.

6 Appendix

The main body of this paper was devoted to showing how the problem of interpreting certain varieties of textured and shaded images can be transformed into equivalent problems in binocular stereo. Beginning with a perspective image, a second (virtual) image is hypothesized according to some presumed model of the original image. The model also specifies how to establish the correspondence between points in the two images. To compute the shape of the surfaces in the original scene, we need only compute the 3-D coordinates from the information in the two images, where the actual scene is a perspective projection and the virtual image has been constructed as an orthographic projection. This appendix shows how three-dimensional coordinates can be computed from point correspondences between a perspective and an orthographic projection when the relation between the imaging geometries is unknown.

We will use lowercase letters to denote image coordinates and uppercase letters for 3-D object coordinates. Unprimed coordinates will refer to the geometry of the perspective image, and primed coordinates to the orthographic image. Let x_1 and x_2 be the image coordinates of a point in the perspective image relative to an arbitrarily selected origin. Let $-d_1$ and $-d_2$ be the [unknown] image coordinates of the principal point and let f [> 0] be the focal length. The object coordinates associated with an image point are (X_1, X_2, X_3), where the origin coincides with the center of projection and the X_3 axis is perpendicular to the image plane. The X_3 coordinates of any object point will necessarily be positive.

The imaging geometry is as depicted in Figure 13 and yields the following standard perspective equations:

$$x_1 + d_1 = f\frac{X_1}{X3}; \qquad x_2 + d_2 = f\frac{X_2}{X_3} \tag{3}$$

For the orthographic image, x'_1 and x'_2 are the image coordinates (relative to an arbitrary origin) and (X'_1, X'_2, X'_3) is the world coordinate system defined such that

$$x'_1 = X'_1; \qquad x'_2 = X'_2 \tag{4}$$

We use the unknown scale factor between orthographic image coordinates and the scene as our unit of measurement.

The two world coordinate systems can be related as follows:

$$X' = R(X - T) \qquad , \tag{5}$$

where X is the column vector $[X_1,\ X_2,\ X_3]^T$,
X' is the column vector $[X'_1,\ X'_2,\ X'_3]^T$,

R is a 3x3 rotation matrix, and
T is a translation vector from the center of perspective projection to the origin of the world coordinate system associated with the orthographic projection. For either component (i=1 or 2), we can write

$$X'_i = R_i \cdot (X - T) \tag{6}$$

where R_i is the i-th row of R. By substituting Equations 3 and 4 into the above, we obtain

$$X_3 = \frac{f(x'_1 + R_1 \cdot T)}{R_1 \cdot [(x_1 + d_1), \ \ (x_2 + d_2), \ \ f]} \tag{7}$$

Eliminating X_3 from the two equations in Equation 7 yields

$$\begin{aligned} 0 = \ & x'_1 x_1 R_{21} + x'_1 x_2 R_{22} + x'_1 R_2 \cdot D - x'_2 x_1 R_{11} - x'_2 x_2 R_{12} - x'_2 R_1 \cdot D \\ & + x_1(R_{21} R_1 \cdot T - R_{11} R_2 \cdot T) + x_2(R_{22} R_1 \cdot T - R_{12} R_2 \cdot T) \\ & + R_1 \cdot T R_2 \cdot D - R_2 \cdot T R_1 \cdot D \end{aligned} \tag{8}$$

where D is the vector $[d_1, \ \ d_2, \ \ f]$.

The above equation relates image coordinates for corresponding points in both images. The following unknowns can be found by using eight corresponding pairs and solving the system of eight linear equations:

$$\begin{aligned} B_1 &= \tfrac{R_{21}}{R_{11}} \\ B_2 &= \tfrac{R_{22}}{R_{11}} \\ B_3 &= \tfrac{R_2 \cdot D}{R_{11}} \\ B_4 &= \tfrac{R_{12}}{R_{11}} \\ B_5 &= \tfrac{R_1 \cdot D}{R_{11}} \\ B_6 &= \tfrac{R_{21}}{R_{11}} R_1 \cdot T - R_2 \cdot T \\ B_7 &= \tfrac{R_{22}}{R_{11}} R_1 \cdot T - \tfrac{R_{12}}{R_{11}} R_2 \cdot T \\ B_8 &= \tfrac{1}{R_{11}}(R_1 \cdot T)(R_2 \cdot D) - \tfrac{1}{R_{11}}(R_2 \cdot T)(R_1 \cdot D) \end{aligned} \tag{9}$$

$$\begin{bmatrix} x'_1x_1 & x'_1x_2 & x'_1 & -x'_2x_2 & -x'_2 & x_1 & x_2 & 1 \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \end{bmatrix} \begin{bmatrix} B_1 \\ B_2 \\ B_3 \\ B_4 \\ B_5 \\ B_6 \\ B_7 \\ B_8 \end{bmatrix} = \begin{bmatrix} x'_2x_1 \\ \cdot \\ \cdot \\ \cdot \end{bmatrix} \tag{10}$$

When more than eight points are available, a least-squares method can be employed to solve the system of equations. Once we have the B_i's in hand, we can solve for the components of the rotation matrix R. First, R_{11} can be determined by making use of the fact that the rows of a rotation matrix are orthogonal. Thus, from $R_1 \cdot R_2 = 0$ and the expressions for B_1, B_2 and B_4 in Equation 9, we get

$$R_{11}^4(B_4^2B_1^2 + B_2^2 - 2B_1B_2B_4) - R_{11}^2(1 + B_1^2 + B_2^2 + B_4^2) + 1 = 0 \tag{11}$$

This yields two real values for R_{11}; fortunately we'll be able to identify the incorrect one later. For now, let us simply choose one at random and return to this point if it turns out to be wrong.

The rest of R can be derived from the B_i's in a similar fashion. R_{12}, R_{21} and R_{22} can be established immediately from R_{11} and Equation 9. R_{13} is determined from the fact that $\| R_1 \| = 1$. $R_1 \cdot R_2 = 0$ gives an expression for R_{23}. Finally, R_3 is computed from the fact that $R_1 \times R_2 = R_3$ for all rotation matrices. As a result, we have completely derived two alternative R matrices, depending on the choice of R_{11}. One of these matrices is correct, while the other can be eliminated later.

Now to solve for the translation vector T. First let us note that T cannot be found uniquely, because the origin of the primed world coordinate system has not been completely specified. The X'_1 and X'_2 coordinates of the origin were fixed by the choice of origin for the orthographic image coordinates, but the position of the origin along the X'_3 axis is still unconstrained. Since we are free to choose any origin for X', we will choose the one for which $T_3 = 0$.

Using the expression for B_6 in Equation 9, we find

$$B_6 = \frac{R_{21}}{R_{11}}(R_{11}T_1 + R_{12}T_2 + R_{13}T_3) - (R_{21}T_1 + R_{22}T_2 + R_{23}T_3) \qquad (12)$$

Making use of the fact that $R_{33} = R_{11}R_{22} - R_{12}R_{21}$ and $T_3 = 0$, we get

$$T_2 = -B_6\frac{R_{11}}{R_{33}} \qquad (13)$$

Similarly,

$$T_1 = B_7\frac{R_{11}}{R_{33}} \qquad (14)$$

The origin of the primed coordinate system in unprimed coordinates is given by

$$T = [B_7\frac{R_{11}}{R_{33}},\ \ -B_6\frac{R_{11}}{R_{33}},\ \ 0]. \qquad (15)$$

If the location of the principal point is known but the focal length (the scale factor of the perspective image) is not, f can easily be computed from Equation 9:

$$f = \frac{B_5R_{11} - R_{11}d_1 - R_{12}d_2}{R_{13}} \qquad (16)$$

If the focal length is known, the principal point of the perspective image is found as follows. Use the third and fifth expressions of Equation 9 to write two equations in the two unknowns, d_1 and d_2. Their solution yields

$$\begin{aligned} d_1 &= f\tfrac{R_{31}}{R_{33}} + \tfrac{B_5R_{11}R_{22} - B_3R_{11}R_{12}}{R_{33}} \\ d_2 &= f\tfrac{R_{32}}{R_{33}} + \tfrac{B_3R_{11}^2 - B_6R_{11}R_{21}}{R_{33}} \end{aligned} \qquad (17)$$

The perspective image coordinates of the principal point are $[-d_1,\ \ -d_2]$.

If neither the focal length nor principal point is known beforehand, then the problem we have proposed does not have a unique solution. Equation 17 specifies the constraints between focal length and piercing point. For any choice of focal length, there exists a unique principal point. The center of perspective projection is constrained to lie on a line parallel to the lines of sight of the orthographic projection. The reconstructed surface will be distorted as one varies the center of projection along this line. It is worth noting, however, that our computations of the rotation matrix R and the translation vector T did not require knowledge of either the focal length or the principal point.

We are now in a position to compute the world coordinates of all points for which we have correspondences. There may, of course, be many more than the minimum of eight points used so far. Equation 6 gives

$$x'_1 = R_1 \cdot [\frac{X_3}{f}(x_1 + d_1), \ \ \frac{X_3}{f}(x_2 + d_2), \ \ X_3] - R_1 \cdot T \tag{18}$$

which can be solved for

$$X_3 = \frac{f(x'_1 + R_1 \cdot T)}{R_{11}x_1 + R_{12}x_2 + R_1 \cdot D} \tag{19}$$

Now we must check the signs of the X_3's. If they are negative, the world points are located behind the center of projection. The correct solution, corresponding to all positive values of X_3, can be found by choosing the alternative value of R_{11} derived earlier and repeating the computations from that point. After obtaining the set of positive X_3's, we can continue.

Equation 3 gives the other unprimed world coordinates:

$$X_1 = \frac{X_3}{f}(x_1 + d_1); \qquad X_2 = \frac{X_3}{f}(x_2 + d_2) \tag{20}$$

If desired, the primed coordinates can be found by applying Equation 5.

The above derivation makes the implicit assumption that the X'_1 and X'_2 axes are scaled equally. It is conceivable that the virtual image coordinates could be unequally scaled, as is the case when they are derived from a rectangular grid (*e.g.*, Figure 4). If we have prior knowledge of the ratio of the sides of each rectangular grid element, then the virtual image coordinates should be normalized before applying this algorithm (*i. e.*, by dividing X'_2 by this ratio). Without knowledge of the ratio, the problem is underspecified and a unidimensional class of solutions exists. Knowledge of the piercing point, if available, can be used to constrain the problem further and to solve for the unique solution. To do this, we define the following virtual coordinate systems in place of Equation (4):

$$x'_1 = X'_1; \qquad x'_2 = \frac{1}{k}X'_2 \tag{21}$$

where k is the ratio of the sides of the rectangular grid elements.

The solution proceeds as before, yielding

$$\begin{aligned}0 = \; & x_1'x_1R_{21} + x_1'x_2R_{22} + x_1'R_2 \cdot D - kx_2'x_1R_{11} - kx_2'x_2R_{12} - kx_2'R_1 \cdot D \\ & +x_1(R_{21}R_1 \cdot T - R_{11}R_2 \cdot T) + x_2(R_{22}R_1 \cdot T - R_{12}R_2 \cdot T) \\ & +R_1 \cdot TR_2 \cdot D - R_2 \cdot TR_1 \cdot D\end{aligned} \tag{22}$$

The above equation is recast as the eight linear equations:

$$\begin{bmatrix} -x_1'x_2 & -x_1' & x_2'x_1 & x_2'x_2 & x_2' & x_1 & x_2 & 1 \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \end{bmatrix} \begin{bmatrix} C_1 \\ C_2 \\ C_3 \\ C_4 \\ C_5 \\ C_6 \\ C_7 \\ C_8 \end{bmatrix} = \begin{bmatrix} x_1'x_1 \\ \cdot \\ \cdot \\ \cdot \end{bmatrix} \tag{23}$$

where

$$\begin{aligned} C_1 &= \tfrac{R_{22}}{R_{21}} \\ C_2 &= \tfrac{R_2 \cdot D}{R_{21}} \\ C_3 &= \tfrac{kR_{11}}{R_{21}} \\ C_4 &= \tfrac{kR_{12}}{R_{21}} \\ C_5 &= \tfrac{kR_1 \cdot D}{R_{21}} \\ C_6 &= \tfrac{R_{11}}{R_{21}}R_2 \cdot T - R_1 \cdot T \\ C_7 &= \tfrac{R_{12}}{R_{21}}R_2 \cdot T - \tfrac{R_{22}}{R_{21}}R_1 \cdot T \\ C_8 &= \tfrac{1}{R_{21}}(R_2 \cdot T)(R_1 \cdot D) - \tfrac{1}{R_{21}}(R_1 \cdot T)(R_2 \cdot D) \end{aligned} \tag{24}$$

The following equalities can then be derived from Equation (24):

$$\begin{aligned} R_{13} &= \tfrac{R_{21}}{fk}(C_5 - C_3d_1 - C_4d_2) \\ R_{23} &= \tfrac{R_{21}}{f}(C_2 - d_1 - C_1d_2) \end{aligned} \tag{25}$$

$$f = \sqrt{\frac{-(C_5 - C_3d_1 - C_4d_2)(C_2 - d_1 - C_1d_2)}{C_3 + C_1C_4}} \tag{26}$$

$$R_{21} = \pm \frac{f}{\sqrt{f^2 + C_1^2 f^2 + (C_2 - d_1 - C_1d_2)^2}} \tag{27}$$

$$k = \frac{R_{21}}{f}\sqrt{f^2C_3^2 + f^2C_4^2 + (C_5 - C_3d_1 - C_4d_2)^2} \qquad (28)$$

The rest of R can now be computed easily from Equation (24) and $R_1 \times R_2 = R_3$. The translation vector T is given by Equation (15) because $C_6 = B_6$ and $C_7 = B_7$. With R and T now fully recovered, it is a simple matter to derive the object coordinates from Eqs. (3), (21), and (5). Let us recall that we have two candidate matrices R hinging on the choice for R_{21}; as before, the correct one must be selected by examining the signs of the X_3 coordinates.

To summarize, we have described an algorithm to compute the relative orientation and position between two imaging systems—perspective and orthographic—from the locations of eight (or more) corresponding image points. Either the principal point or the focal length and rectangular aspect ratio are computed along the way. With this information in hand, the world coordinates of all points in the imaged scene can be computed.

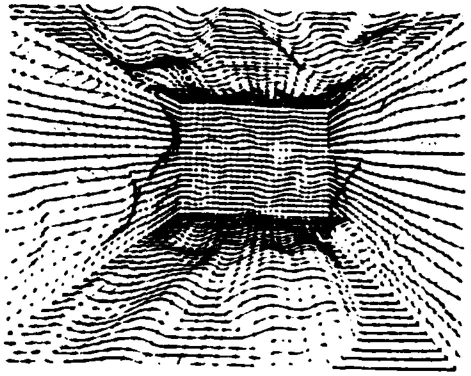


Figure 1: Wire Room

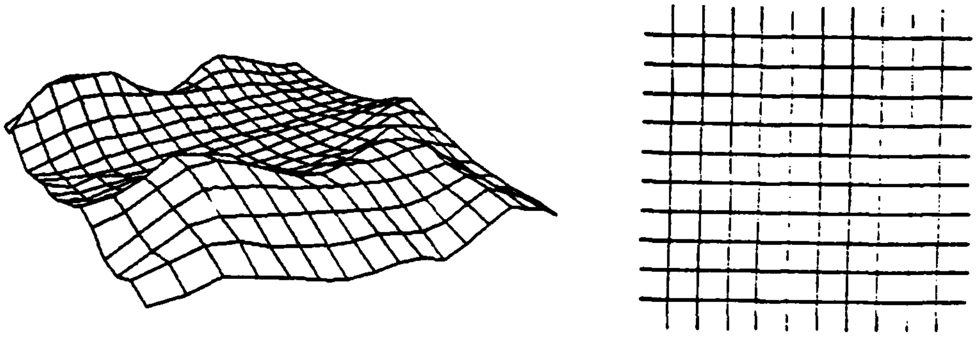


Figure 2: (a) A projected texture (b) Its virtual image

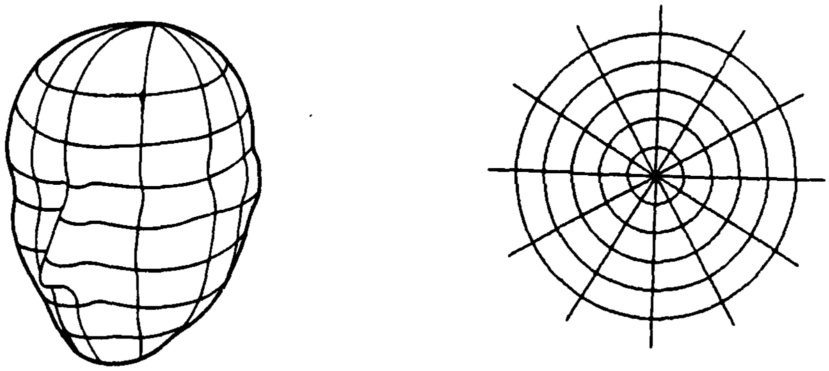


Figure 3: (a) The original image (b) The virtual image

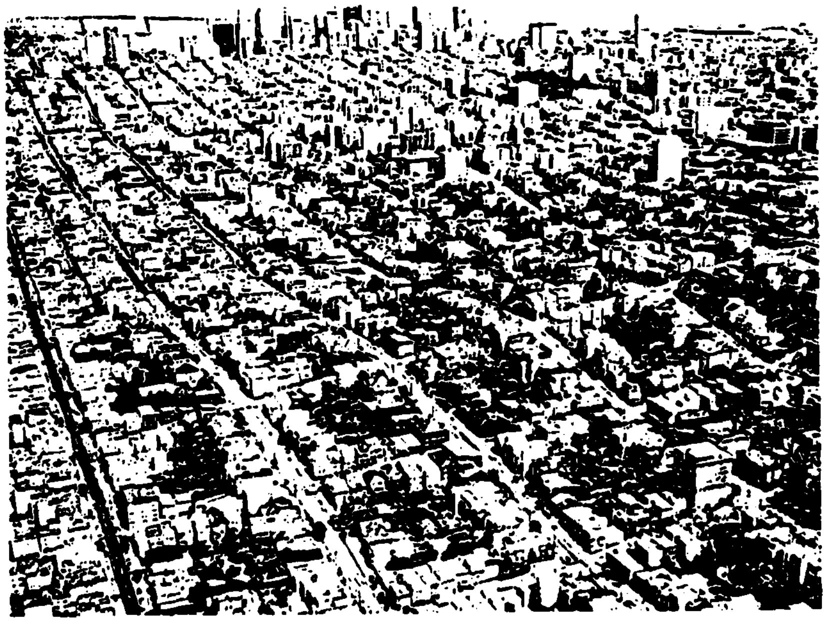


Figure 4: The streets in this scene form a natural projected texture.

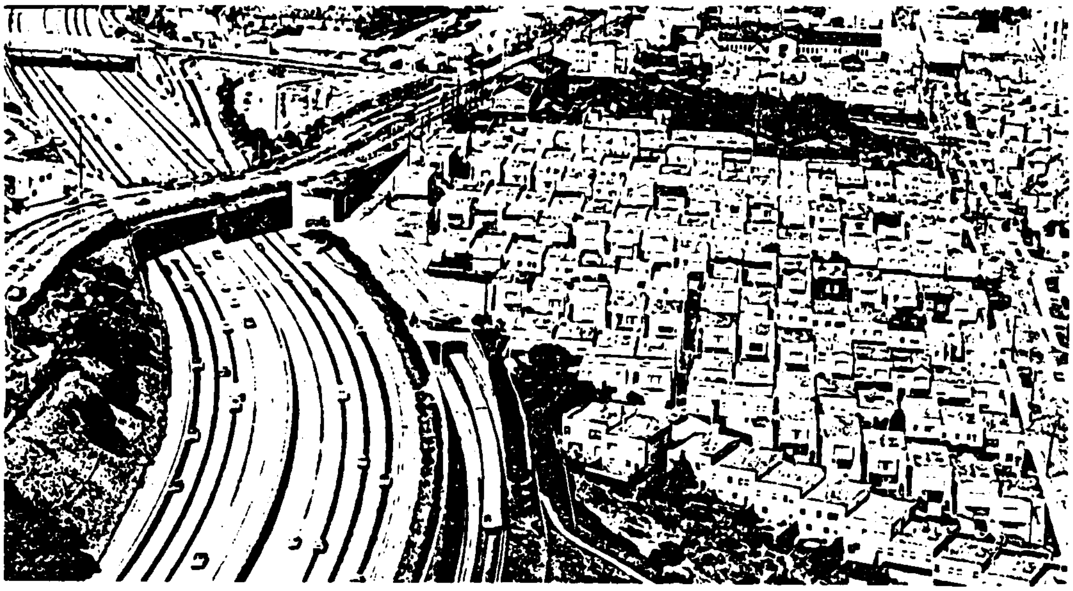


Figure 5: The houses can be construed as a projected texture.

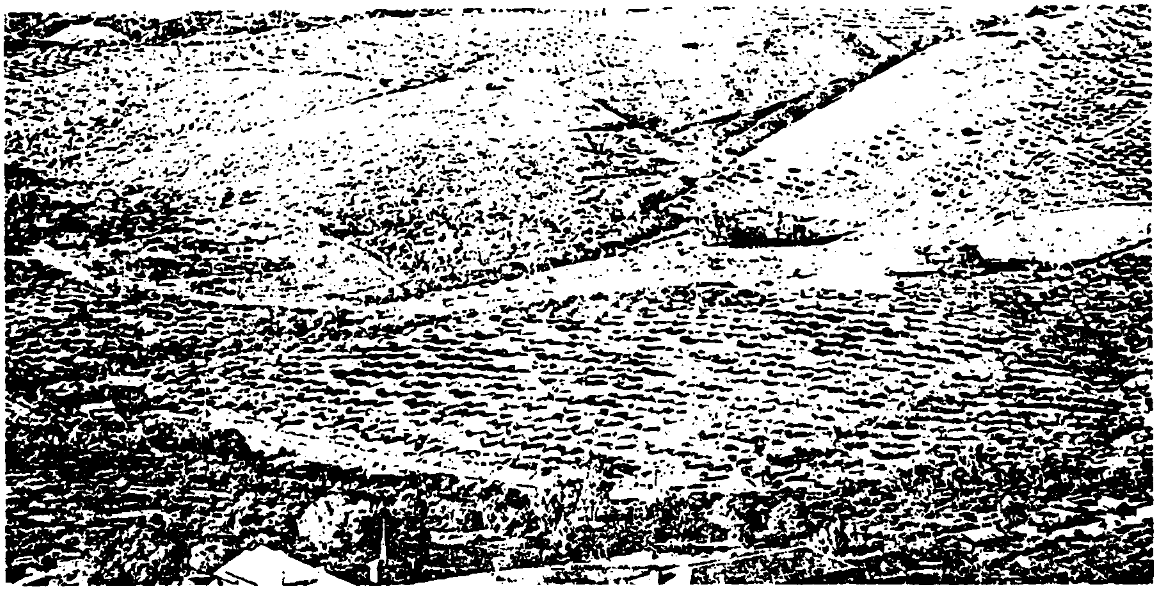


Figure 6: These grapevines exhibit a regular texture.

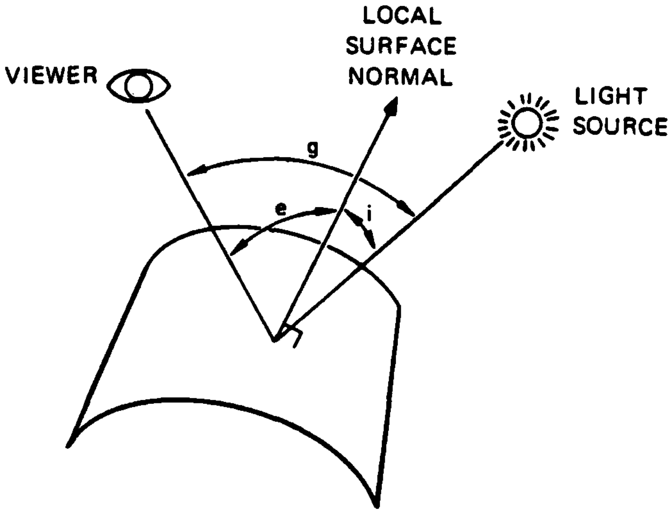


Figure 7: The geometry of surface illumination

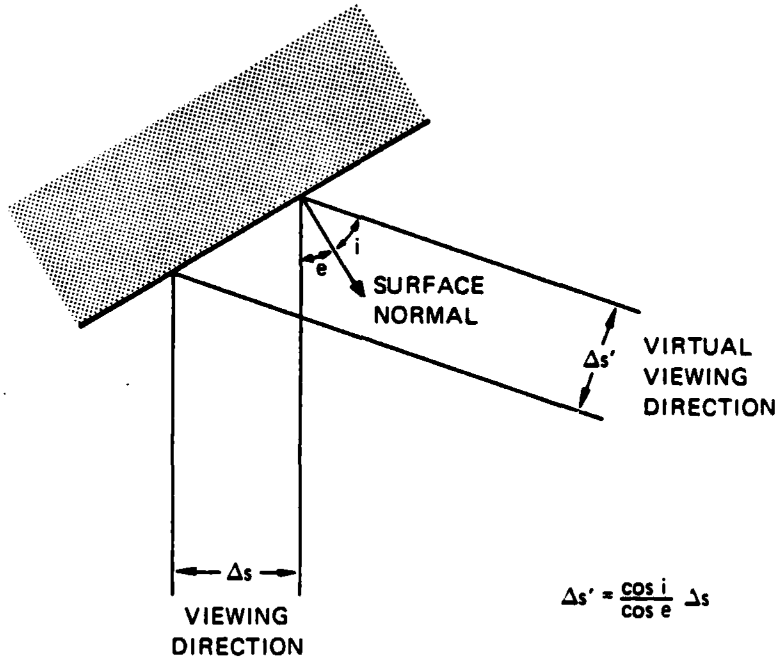

i
e
SURFACE NORMAL
Δs'
VIRTUAL VIEWING DIRECTION
Δs
VIEWING DIRECTION
$\Delta s' = \frac{\cos i}{\cos e} \Delta s$

Figure 8: The geometry along a line in the direction of the light source

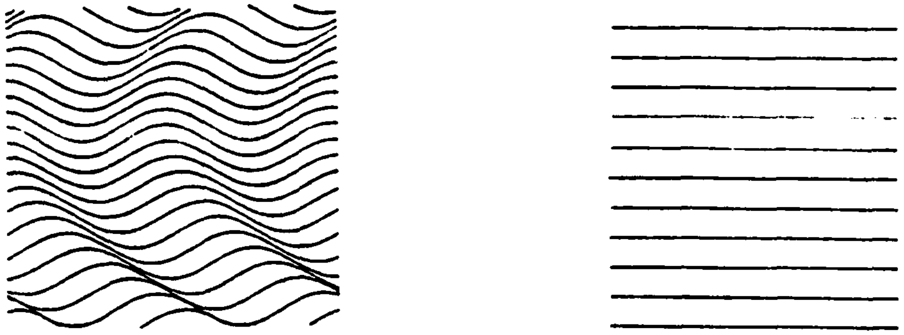


Figure 9: (a) An image of contours (b) Its virtual image

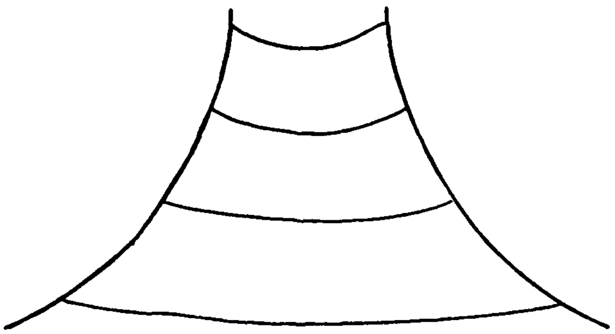


Figure 10: This simple drawing has two reasonable interpretations. It is seen as curved roller-coaster tracks if the lines are assumed to be the projection of a rectangular grid, or as a volcano when the lines are assumed to be the projection of a circular grid.

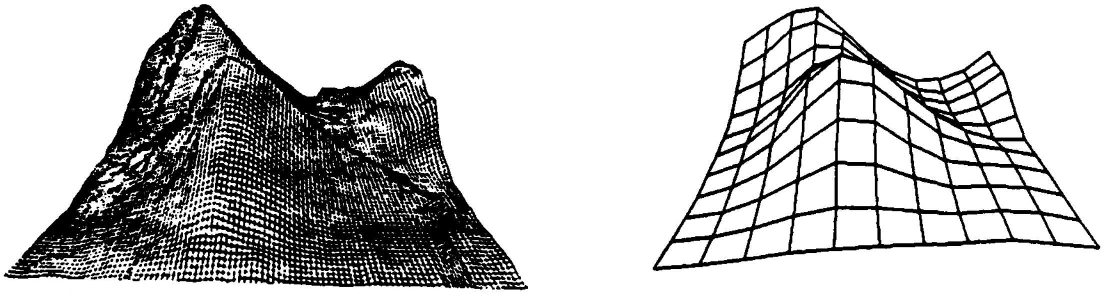


Figure 11: (a) View of part of a DTM (b) View of surface reconstructed from (a)

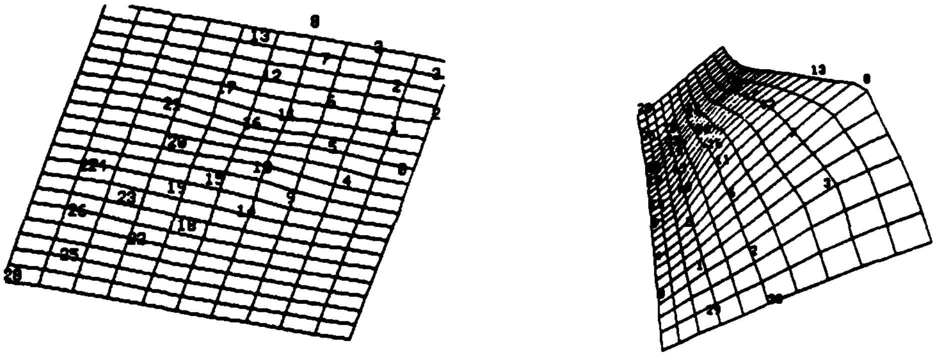
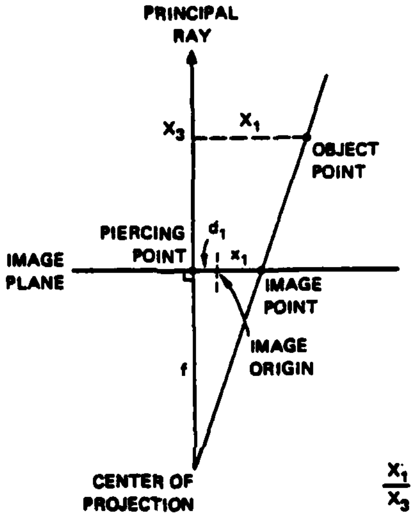


Figure 12: (a) Orthographic view of surface reconstructed from Figure 4 (b) Perspective view of same surface (from derived camera location)



$$\frac{X_1}{X_3} = \frac{x_1 + d_1}{f}$$

Figure 13: Definition of coordinate system

Appendix D

A New Sense for Depth of Field

By: Alex P. Pentland

A NEW SENSE FOR DEPTH OF FIELD

Alex P. Pentland

Artificial Intelligence Center, SRI International
333 Ravenswood Ave, Menlo Park, CA 94025
and
Center for the Study of Language and Information
Stanford University, Stanford CA 94038

ABSTRACT

One of the major unsolved problems in designing an autonomous agent [robot] that must function in a complex, moving environment is obtaining reliable, real-time depth information, preferably without the limitations of active scanners. Stereo remains computationally intensive and prone to severe errors, the use of motion information is still quite experimental, and autofocus schemes can measure depth at only one point at a time. We examine a novel source of depth information: focal gradients resulting from the limited depth of field inherent in most optical systems. We prove that this source of information can be used to make reliable depth maps of useful accuracy with relatively minimal computation. Experiments with realistic imagery show that measurement of these optical gradients can potentially provide depth information roughly comparable to stereo disparity or motion parallax, while avoiding image-to-image matching problems. A potentially real-time version of this algorithm is described.

I. INTRODUCTION

Our subjective impression is that we view our surroundings in sharp, clear focus. This impression is reinforced by the virtually universal photographic tradition[**] to make images that are everywhere in focus, i.e., that have infinite depth of field. Unfortunately, both this photographic tradition and our feeling of a sharply focused world seems to have lead vision researchers — in both human and machine vision — to largely ignore the fact that in biological systems the images that fall on the retina are typically quite *badly* focused everywhere except within the central fovea [1,2]. There is a *gradient* of focus, ranging from nearly perfect focus at the point of regard to almost complete blur at points on distant objects.

It is puzzling that biological visual systems first employ an optical system that produces a degraded image, and then go to great lengths to undo this blurring and present us with a subjective impression of sharp focus. This is especially peculiar because it is just as easy to start out with everything in perfect focus. Why, then, does Nature prefer to employ a lens system in which most of the image is blurred?

In this paper we report the finding that this gradient of focus inherent in biological and most other optical systems is a useful source of depth information, prove that these focal gradients may be used to recover a depth map (i.e., distances between viewer and points in the scene) by means of a few, *simple transformations of the image*, and that with additional computation the reliability of this depth information may be internally checked. This source of depth information (which differs markedly from that used in automatic focusing methods) has not previously been described in the human vision literature, and we have been unable to find any investigation of it in the somewhat more scattered machine vision literature. The performance of a practical technique has been demonstrated on realistic imagery, and an inexpensive, real-time version of the algorithm is described. Finally, we report experiments showing that people make significant use of this depth information.

This novel method of obtaining a depth map is important because there is currently no passive sensing method for obtaining depth information that is simultaneously fast enough, reliable enough, and produces a sufficiently dense depth map to support the requirements of a robot moving in a complex environment. Stereopsis, despite huge investment, remains computationally intensive and prone to severe errors, the use of motion information is still in an experimental stage, and autofocus schemes can measure depth at only one point at a time. We believe that this research, therefore, will prove a significant advance in solving the problem of real-time acquisition of reliable depth maps without the limitations inherent in active scanners (e.g., laser rangefinders).

II. THE FOCAL GRADIENT

Most biological lens systems are exactly focused[*] at only one distance along each radius from the lens into the scene. The locus of exactly focused points forms a doubly curved, approximately spherical surface in three-dimensional space. Only when objects in the scene intersect this surface is their image exactly in focus; objects distant from this surface of exact focus are blurred, an effect familiar to photographers as depth of field.

The amount of defocus or blurring depends solely on the distance to the surface of exact focus and the characteristics of the lens system; as the distance between the imaged point and the surface of exact focus increases, the imaged objects become progressively more defocused. If we could measure the amount of blurring at a given point in the image, therefore, it seems possible that we could use our knowledge of the parameters of the lens system to compute the distance to the corresponding point in the scene.

[*] This research was made possible in part by a grant from the Systems Development Foundation, and by a grant from the National Science Foundation, Grant No. DCR-83-12766, and by Defense Advanced Research Projects Agency contract no. MDA 903-83-C-0027

[**] A practice established in large part by Ansel Adams and others in the famous "f/64 Club"

[*] "Exact focus" is taken here to mean "has the minimum variance point spread function," the phrase "measurement of focus" is taken to mean "characterize the point spread function."

The distance D to an imaged point is related to the parameters of the lens system and the amount of defocus by the following equation, which is developed in the appendix.

$$D = \frac{Fv_0}{v_0 - F - \sigma f} \tag{1}$$

where v_0 is the distance between the lens and the image plane (e.g., the film location in a camera), f the f-number of the lens system, F the focal length of the lens system, and σ the spatial constant of the point spread function (i.e., the radius of the imaged point's "blur circle") which describes how an image point is blurred by the imaging optics. The point spread function may be usefully approximated by a two-dimensional Gaussian $G(r, \sigma)$ with a spatial constant σ and radial distance r. The validity of using a Gaussian to describe the point spread function is discussed in the appendix.

In most situations, the only unknown on the right-hand side of Equation (1) is σ, the point spread function's spatial parameter. Thus, we can use Equation (1) to solve for absolute distance given only that we can measure σ, i.e., the amount of blur at a particular image point.

Measurement of σ presents a problem, however, for the image data is the result of both the characteristics of the scene and those of the lens system. To disentangle these factors, we can either look for places in the image with known characteristics (e.g., sharp edges), or we can observe what happens when we change some aspect of the lens system. In the following discussion both of these two general strategies for measurement of σ are described: the use of sharp edges, and comparison across different aperture settings. Both approaches require only one view of the scene.

A. Using Sharp Discontinuities

Image data are determined both by scene characteristics and the properties of the lens system, e.g., how fast image intensity changes depends upon both how scene radiance changes and the diameter of the blur circle. If we are to measure blur circle, therefore, we must already know the scenes' contribution to the image. At edges — sharp discontinuities in the image formation process — the rate of change we observe in the image is due primarily to the point spread function; because we can often recognize sharp discontinuities with some degree of confidence [3,4] we can use image data surrounding them to determine the focus. These observations lead to the following scheme for recovering the viewer-to-scene* distance at points of discontinuity.

Mathematical Details. To calculate the spatial constant of the point spread function we require a measure of the rate at which image intensity is changing; the wide-spread use of zero-crossings of the Laplacian to find edges [5] suggests using slope of the Laplacian across the zero-crossing as a measure of rate of change.

Consider a vertical step edge in the image of magnitude δ at position x_0. In this case the values $C(x, y)$ resulting from the convolution of image intensities $I(x, y)$ with the Laplacian of a Gaussian $\nabla^2 G(r, \sigma)$ (as in [5]) have the form

$$\begin{aligned} C(x,y) &= \nabla^2 G(r,\sigma) \otimes I(x,y) \\ &= \int\int \nabla^2 G(\sqrt{(x-u)^2 + (y-v)^2}, \sigma) I(u,v)\,du\,dv \\ &= \delta(dG(x - x_0, \sigma)/dx) \end{aligned} \tag{2}$$

where $G(x - x_0, \sigma)$ is a one-dimensional Gaussian centered at point x_0, and σ is the spatial constant of the point spread function at that point in the image. For such an edge the slope of the function $C(x, y)$ at the point of the zero-crossing is equal to the maximum rate of change in image intensity, and so we can use it to estimate σ.

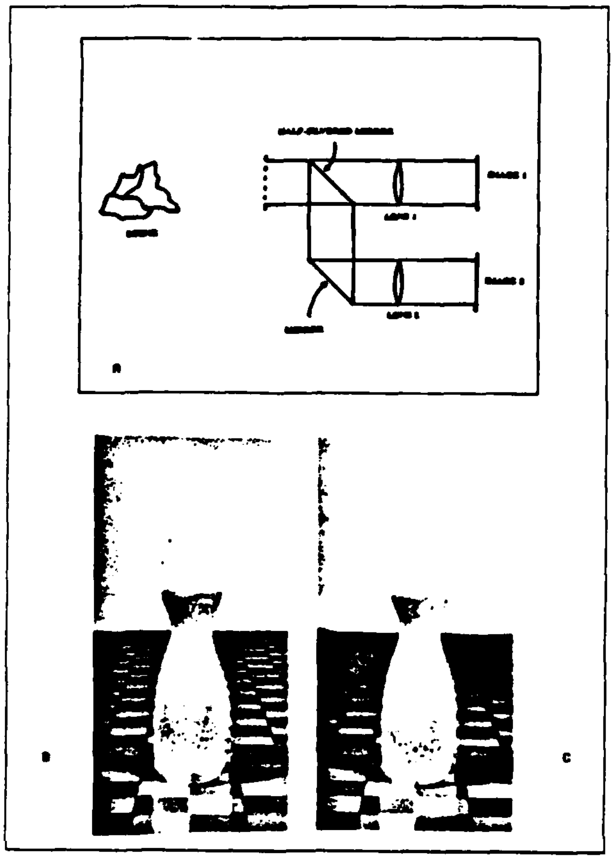


Figure 1. Images Identical Except for Depth of Field. (a) Production: The light from a single view is split into two identical images and directed through two lens systems with different aperture size. Alternatively, one can vary the aperture between alternate frames from a standard video or CCD camera. In either case the two resulting images are identical except for depth of field, as shown in Figure 1 (b) and (c). These images are of a mirrored bottle on a checkered plain.

An estimate of σ can be formed as follows:

$$C(x, y) = \delta \frac{dG(x, \sigma)}{dx} = \frac{-\delta x}{\sqrt{2\pi}\sigma} \exp\left(-\frac{x^2}{2\sigma^2}\right) \tag{3}$$

where x, y and δ are as before, and for convenience x_0 is taken to be zero. Taking the absolute value and then the natural log, we find

$$\ln \frac{\delta}{\sqrt{2\pi}\sigma^3} - \frac{x^2}{2\sigma^2} = \ln\left|\frac{C(x,y)}{x}\right| \tag{4}$$

We can formulate Equation (4) as

$$Ax^2 + B = C \tag{5}$$

where

$$A = -\frac{1}{2\sigma^2} \qquad B = \ln \frac{\delta}{\sqrt{2\pi}\sigma^3} \qquad C = \ln\left|\frac{C(x,y)}{x}\right|$$

If we interpret Equation (5) as a linear regression in x^2 we can then obtain a maximum-likelihood estimate of the constants A and B,

*When the discontinuity is in depth, as at an occluding contour, the distance measured is to the nearer side of the discontinuity.

and thus obtain σ. The solution of this linear regression is

$$A = \frac{\sum_i (x_i^2 - \bar{x}^2)C_i}{\sum_i (x_i^2 - \bar{x}^2)^2} \qquad B = \bar{C} - \bar{x}^2 A \tag{6}$$

where $\bar{x}$ is the mean of the x_i, and $\bar{C}$ is the mean of the C_i. From A in Equation (6) we can obtain the following estimate of the value of the spatial constant σ:

$$\sigma = (-2A)^{-2}$$

Having estimated σ we can now use Equation (1) to find the distance to the imaged point; note that there are two solutions, one corresponding to a point in front of the locus of exact focus, the other corresponding to a point behind it. This ambiguity is generally unimportant because we can usually arrange things so that the surface of exact focus is nearer to the sensor than any of the objects in the field of view.

B. Comparison Across Differing Apertures

The limiting factor in the previous method is the requirement that we must know the scene characteristics before we can measure the focus; this restricts the applicability of the method to special points such as step discontinuities. If, however, we had two images of exactly the same scene, but with different depth of field, we could factor out the contribution of the scene to the two images (as the contribution is the same), and measure the focus directly.

Figure 1 shows one method of taking a single view of the scene and producing two images that are identical except for aperture size and therefore depth of field. This lens system uses a half-silvered mirror (or comparable contrivance) to split the original image into two identical images, which are then directed through lens systems with different aperture size. Because change in aperture does not affect the position of image features, the result is two images that are *identical* except[*] for their *focal* gradient (amount of depth of field), and so there is no difficulty in matching points in one image to points in the other. Figures 1 (b) and (c) show a pair of such images. Alternatively, one could rig a video or CCD camera so that alternate frames employ a different aperture; as long as no significant motion occurs between frames the result will again be two images identical except for depth of field.

Because differing aperture size causes differing focal gradients, the same point will be focused differently in the two images; for our purposes the critical fact is that the magnitude of this difference is a simple function of the distance between the viewer and the imaged point. To obtain an estimate of depth, therefore, we need only compare corresponding points in the two images and measure this change in focus. Because the two images are identical except for aperture size they may be compared directly; i.e., there is no matching problem as there is with stereo or motion algorithms. Thus we can then recover the absolute distance D by simple point-by-point comparison of the two images, as described below.

Mathematical Details. We start by taking a patch $f_1(r,\theta)$ centered at (x_0, y_0) within the first image $I_1(x,y)$:

$$f_1(r,\theta) = I_1(x_0 + r\cos\theta, y_0 + r\sin\theta)$$

and calculate its two-dimensional Fourier transform $\mathcal{F}_1(t,\theta)$. The same is done for a patch $f_2(r,\theta)$ at the corresponding point in the second image, giving us $\mathcal{F}_2(t,\theta)$. Again, note that there is no matching problem, as the images are identical except for depth of field.

Now consider the relation of f_1 to f_2. Both cover the same region in the image, so that if there were no blurring both would be equal to the same intensity function $f_0(r,\theta)$. However, because there is blurring (with spatial constants σ_1 and σ_2), we have

$$\frac{f_1(r,\theta)}{f_2(r,\theta)} = \frac{f_0(r,\theta) \otimes G(r,\sigma_1)}{f_0(r,\theta) \otimes G(r,\sigma_2)} \tag{7}$$

[One point of caution is that Equation (7) may be substantially in error in cases with a large amount of defocus, as points neighboring the patches f_1, f_2 will be "spread out" into the patches by differing amounts. This problem can be minimized by using patches whose edges trail off smoothly, e.g., $f_1(r,\theta) = I(x_0 + r\cos\theta, y_0 + r\sin\theta)G(r,\omega)$ for appropriate spatial parameter ω.]

Noting that

$$f(r,\theta) = e^{-\pi r^2} \qquad \mathcal{F}(\lambda,\theta) = e^{-\pi\lambda^2}$$

are a Fourier pair and that if $f(r,\theta)$ and $\mathcal{F}(\lambda,\theta)$ are a Fourier pair then so are

$$f(\alpha r,\theta) \qquad \frac{1}{|\alpha|}\mathcal{F}(\frac{\lambda}{\alpha},\theta)$$

we see that we may use Equation (7) to derive the following relationship between $\mathcal{F}_1$ and $\mathcal{F}_2$ (the Fourier transforms of image patches f_1 and f_2) and $\mathcal{F}_0$ (the transform of the [hypothetical] unblurred image patch f_0):

$$\mathcal{F}_1(\lambda,\theta) = \frac{\mathcal{F}_0(\lambda,\theta)G(\lambda,\frac{1}{\sqrt{2\pi}\sigma_1})}{\sqrt{2\pi}\sigma_1} \qquad \mathcal{F}_2(\lambda,\theta) = \frac{\mathcal{F}_0(\lambda,\theta)G(\lambda,\frac{1}{\sqrt{2\pi}\sigma_2})}{\sqrt{2\pi}\sigma_2} \tag{8}$$

Thus[*] ,

$$\frac{\mathcal{F}_1(\lambda)}{\mathcal{F}_2(\lambda)} = \frac{G(\lambda,\sigma_1)\sigma_2}{G(\lambda,\sigma_2)\sigma_1} = \frac{\sigma_2^2}{\sigma_1^2}\exp(\lambda^2 2\pi^2(\sigma_2^2 - \sigma_1^2)) \tag{9}$$

where

$$\mathcal{F}(\lambda) = \int_{-\pi}^{\pi} \mathcal{F}(\lambda,\theta)d\theta$$

Thus, given $\mathcal{F}_1$ and $\mathcal{F}_2$ we can find σ_1 and σ_2, as follows. Taking the natural log of Equation (9) we obtain

$$\ln\frac{\sigma_2^2}{\sigma_1^2} + \lambda^2 2\pi^2(\sigma_2^2 - \sigma_1^2) = \ln\mathcal{F}_1(\lambda) - \ln\mathcal{F}_2(\lambda)$$

We may formulate this as $A\lambda^2 + B = C$ where

$$A = 2\pi^2(\sigma_2^2 - \sigma_1^2) \qquad B = \ln\frac{\sigma_2^2}{\sigma_1^2} \qquad C = \ln\mathcal{F}_1(\lambda) - \ln\mathcal{F}_2(\lambda)$$

i.e., as a linear regression equation in λ^2. The solution to this regression equation is the same as shown in the last example, and gives us maximum-likelihood estimates of A and B. Solving A and B for σ_1 and σ_2 yields

$$\sigma_1 = \sqrt{\frac{A}{2\pi^2(e^B - 1)}} \qquad \sigma_2 = \sqrt{\frac{Ae^B}{2\pi^2(e^B - 1)}} \tag{10}$$

We may now use these estimates of σ_1 and σ_2 to calculate absolute distance to the imaged surface patch. Using Equation (1) for each of the two images, we see that we now have

$$D = \frac{Fv_0}{v_0 - F - \sigma_1 f_1} \qquad D = \frac{Fv_0}{v_0 - F - \sigma_2 f_2} \tag{11}$$

where f_1 and f_2 are the f-numbers for the two halves of the imaging system.

[*] Their overall brightness might also differ.

[*] Note that we need only consider the amplitude of the transforms in these calculations.

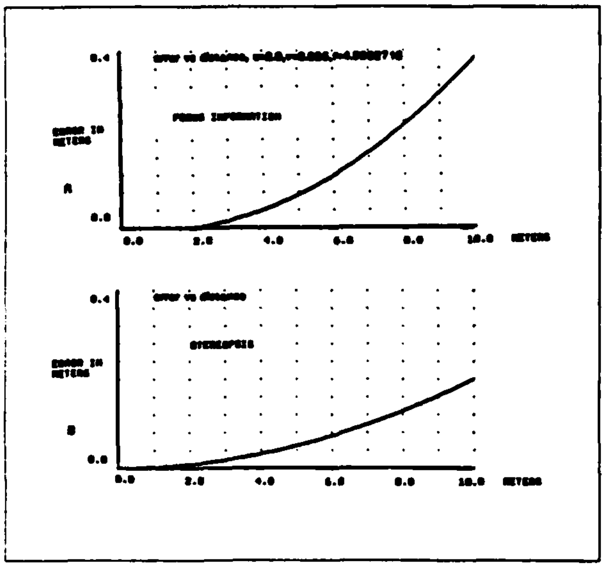


Figure 2. Accuracy at estimating distance, assuming human visual system parameters, using (a) focal gradient information, and (b) stereopsis.

C. Checking the answer: overconstraint

We may solve either of the two equations in (11) for D, the distance to the imaged surface patch. Thus the solution is overconstrained; both solutions must produce the same estimate of distance—otherwise the estimates of σ_1 and σ_2 must be in error. This can occur, for instance, when there is insufficient high-frequency information in the image patch to enable the change in focus to be calculated. The important point is that this overconstraint allows us to check our answer: if the equations disagree, then we know not to trust our answer. If, on the other hand, both equations agree then we can know (to within measurement error) that our answer *must* be correct.

D. Accuracy

Possibly the major question concerning the usefulness of focal gradient information is whether such information can be sufficiently accurate. There are two major issues to be addressed: first, can we estimate the variance σ of the point spread function with sufficient accuracy, and second, does this translate into a reasonable degree of accuracy in the estimation of depth.

Recent research aimed at estimating the point spread function has shown that it may be accurately recovered from unfamiliar images despite the presence of normal image noise [6,7]. Further, it appears that humans can estimate the width of the point spread function to within a few percent [8,9]. These findings, together with the results of estimating σ reported in the next section, show that accurate estimation of σ is practical given sufficient image resolution.

The second issue is whether the available accuracy at estimating σ translates into a reasonable accuracy in estimating depth. Figure 2 (a) show the theoretical error curve for the human eye, assuming the accuracy at estimating σ reported in [4]. It can be seen that reasonable accuracy is available out to several meters. This curve should be compared to the accuracy curve for stereopsis, shown in Figure 2 (b), again assuming human parameters. It can be seen that the accuracies are comparable.

E. Human Perception

We have recently reported evidence demonstrating that people make use of the depth information contained in focal gradients [9]; interestingly, the ecological salience of this optical gradient does not appear to have been previously reported in the scientific literature. The hypothesis that the human visual system makes significant use of this cue to depth has been investigated in two experiments.

In the first experiment, pictures of naturalistic scenes were presented with various magnitude of focal gradient information. It was found that increasing the magnitude of the focal gradient results in increasing subjective depth. In the second experiment, subjects were shown a rightward rotating wireframe (Nekker) cube displayed in perspective on a CRT. Such a display may be perceived as either as a rigid object rotating to the right, or (surprisingly) as wobbling, non-rigid object rotating to the left. Normally subjects see the rigid interpretations most of the time, but when we introduced a focal gradient that favored the non-rigid interpretations, the non-rigid interpretations was seen almost as often as the rigid one.

An experiment demonstrating the importance of depth of field in human perception can be easily performed by the reader. First make a pinhole camera by poking a small, clean hole through a piece of stiff paper or metal. Imposition of a pinhole in the line of sight causes the depth of field to be very large, thus effectively removing this depth cue from the image. Close one eye and view the world through the pinhole, holding it as close as possible to the surface of your eye, and note your impression of depth (for those of you with glasses, things will look sharper if you are doing it correctly). Now quickly remove the pinhole and view the world normally (still using only one eye). The change in the sense of depth is remarkable; many observers report that the change is nearly comparable to the difference between monocular and binocular viewing, or the change which occurs when a stationary object begins to move.

III. IMPLEMENTATION AND EVALUATION

A. Using sharp edges

The first method of deriving depth from the focal gradient, by measuring apparent blur near sharp discontinuities, was implemented in a straightforward manner (convolution values near zero-crossings were employed in Equations (4) - (6)) and evaluated on the image shown in Figure 3. In this image the optical system had a smaller depth of field than is currently typical in vision research; this was done because the algorithm requires that the digitization adequately resolve the point spread function.

Figure 3 also shows the depth estimates which were obtained when the algorithm was applied to this image. Part (a) of this Figure 3 shows all the sharp discontinuities identified [2]. It was found that there was considerable variability in the depth estimates obtained along these contours, perhaps resulting from the substantial noise (3 of 8 bits) which was present in the digitized image values. To minimize this variability the zero-crossing contours were segmented at points of high curvature, and the depth values were averaged within the zero-crossing segments. Figures 3 (b), (c), and (d) show the zero-crossing segments that have large, medium, and small depth values, respectively. It can be seen that the image is properly segmented with respect to depth, with the exception of one small segment near the top of (c). This example demonstrates that this depth estimation technique — which requires little computation beyond the calculation of zero-crossings — can be employed to order sharp edges by their depth values.

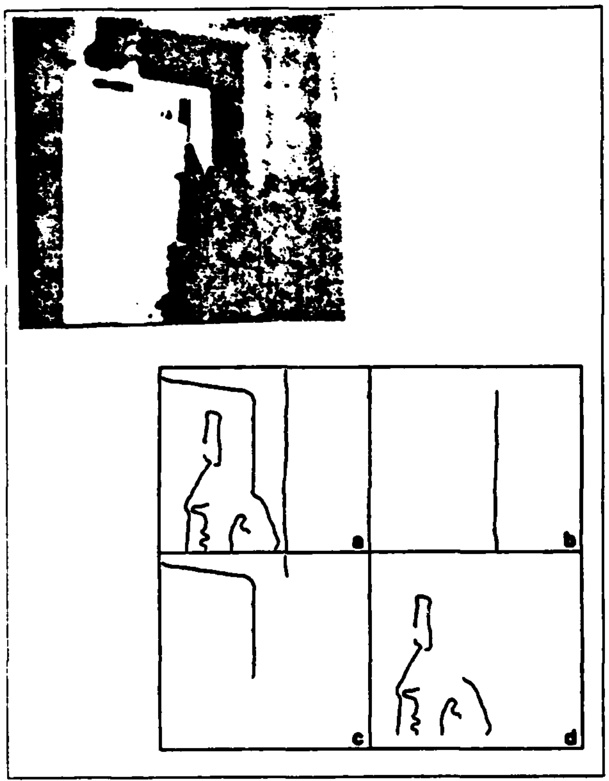


Figure 3. An Indoor Image of a Sand Castle, Refrigerator, and Door, Together with Depth Estimates for its Zero-Crossing Segments. Part (a) of this figure shows all the sharp discontinuities found. Parts (b), (c), and (d) show the zero-crossing segments that have large, medium, and small depth values, respectively. It can be seen that the image is properly segmented with respect to depth, with the exception of one small segment near the top of (c).

B. Comparison of different apertures

The second technique, comparing two images identical except for aperture, can be implemented in many different ways. We will report a very simple version of the algorithm that is amenable to an inexpensive real-time implementation.

In this algorithm two images are acquired as shown in Figure 1 (a); they are identical except for their depth of field and thus the amount of focal gradient present, as shown in Figures 1 (b) and (c). These images are then convolved with a small Laplacian filter, providing an estimate of their local high-frequency content. The output of the Laplacian filters are then summed over a small area and normalized by dividing them by the mean local image brightness, obtained by convolving the original images with a Gaussian filter. It appears that a region as small as 4 x 4 pixels is sufficient to obtain stable estimates of high-frequency content. Figures 4 (a) and (b) show the normalized high-frequency content of Figures 1 (b) and (c).

Finally, the estimated high-frequency content of the blurry, large-aperture image is divided by that of the sharp, small-aperture image, i.e., each point of Figure 4 (a) is divided by the corresponding point in Figure 4(b). This produces a "focal disparity" map, analogous to a stereo disparity map, that measures the change in focus between the two images and whose values are monotonically related to depth by Equation (1). Figure 4 (c) shows the disparity map produced from Figures 2 (b) and 2 (c); intensity in this figure is proportional to depth.

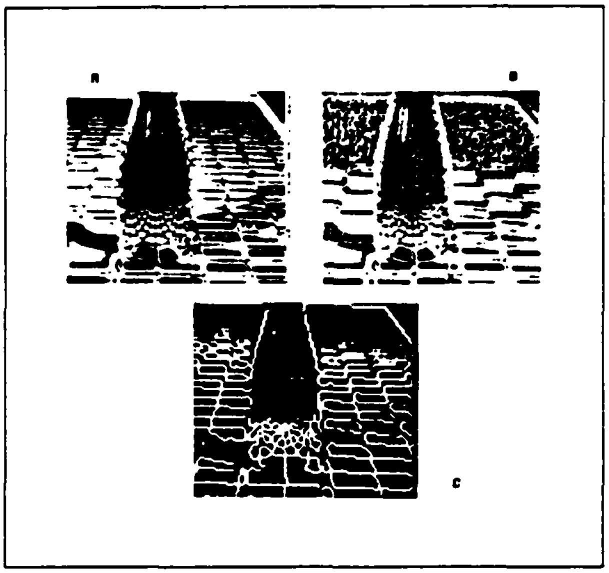


Figure 4. (a) and (b) show the normalized high-frequency content of Figures 2 (b) and (c), respectively. (c) shows the focal disparity map (analogous to a stereo disparity map) obtained by comparing (a) and (b); brightness is proportional to depth.

Areas of 4 (c) that are black have insufficient high-frequency energy in the sharp-focus image to make an estimate of depth.

It can be seen that this disparity map is fairly accurate. Note that points reflected in the bottle are estimated as further than points along the edge of the bottle; this is not a mistake, for these points the distance traveled by the light is further than for those along the edge of the bottle. This algorithm, in common with stereo and motion algorithms, does not "know" about mirrored surfaces.

C. Design for a real-time implementation

A minimum of one convolution per image is required for this technique, together with a left shift and four subtractions for the Laplacian, and three divides for the normalization and comparison. If special convolution hardware is available, one can use two convolutions — one Laplacian and one Gaussian - per image, leaving only three divides* for the normalization and comparison. Frame buffers that can convolve image data in parallel with image acquisition are now available at a reasonable price, leaving as few as 3 operations per pixel to calculate the disparity map. For a 256 x 256 image, this can be accomplished in as little as 0.35 seconds with currently available microcomputers.

IV. DISCUSSION

The most striking aspect of this source of depth information is that absolute range can be estimated from a single view with no image-to-image matching problem, perhaps the major source of error in stereo and motion algorithms. Furthermore, no special scene characteristics need be assumed, so that the techniques utilizing this cue to depth can be generally applicable. The second most striking fact is the simplicity of these algorithms: it appears that a real-time implementation can be accomplished relatively cheaply.

Measurement of the focal gradients associated with limited depth of field appears to be capable of producing depth estimates that are at least roughly comparable to edge- or feature-based stereo and motion

*which can be done by table lookup.

algorithms. The mathematics of the aperture-comparison technique shows it to be potentially more reliable than stereo or motion — i.e., there is no correspondence problem, and one can obtain an internal check on the answer — although (as discussed above) it has somewhat less accuracy.

The sharp-edge algorithm appears to have potential for useful depth-plane segmentation, although it is probably not accurate enough to produce a depth map. I believe that this algorithm will be of some interest because most of the work — finding and measuring the slope of zero-crossings — is often already being done for other purposes. Thus this type of depth-plane segmentation can be done almost as a side effect of edge finding or other operations.

The aperture-comparison algorithm provides considerably stronger information about the scene because it *overconstrains* scene depth, allowing an internal check on the algorithm's answer. Thus it provides depth information with a reliability comparable to the best that is theoretically available from three-or-more image stereo and motion algorithms, although it has somewhat less depth resolution. The major limitation in measuring focal gradient depth information in this manner appears to be insuring sufficient high-frequency information to measure the change between images; this requires having both adequate image resolution and high-frequency scene content.

Summary. In summary, we have described a new source of depth information — the focal gradient — that can provide depth information at least roughly comparable to stereo disparity or motion parallax, while avoiding the image-to-image matching problems that have made stereo and motion algorithms unreliable. We have shown that the limited depth of field inherent in most optical systems can be used to make depth maps of useful accuracy with relatively minimal computation, and have successfully demonstrated a potentially real-time technique for recovering depth maps from realistic imagery. It is our hope, therefore, that this research will prove to be a substantial advance towards building a robot that can function in complex, moving natural environments.

REFERENCES

[1] H. Crane, *A Theoretical Analysis of the Visual Accommodation System in Humans*, Final Report NAS 2-2760, NASA Ames Research Center (1966).

[2] M. Born and E. Wolf, *Principles of Optics*, Pergamon, London (1965).

[3] A. Pentland, *The Visual Inference of Shape: Computation from Local Features*, Ph.D. thesis, Massachusetts Institute of Technology (1982).

[4] A. Witkin, *Intensity-Based Edge Classification*, Proceedings of the American Association for Artificial Intelligence, August 1982, Pittsburgh, Penn.

[5] E. Hildreth, *Implementation of a Theory of Edge Detection*, M.I.T. AI Laboratory Technical Report 579 (April 1980).

[6] K.T. Knox and B.J. Thomson, *Recovery of Images from Atmospherically Degraded Short-Exposure Photographs*, Astrophys. J. 193, L45-L48 (1974).

[7] J.B. Morton and H.C. Andrews, *A Posteriori Method of Image Restoration*, Opt. Soc. Am. 69, 2 (1979) 280-290.

[8] A. Pentland, *Uniform Extrafoveal Sensitivity To Pattern Differences*, Journal of the Optical Society of America, November 1978.

[9] A. Pentland, *The Focal Gradient: Optics Ecologically Salient*, Supplement to Investigative Opthomology and Visual Science, April 1985.

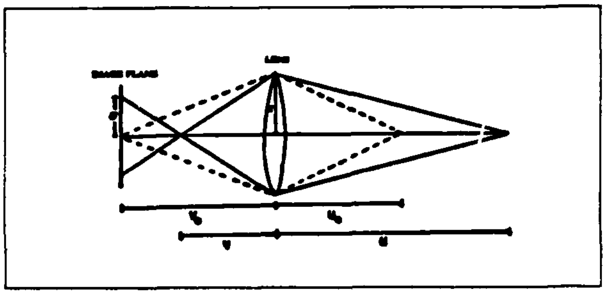


Figure 5. Geometry of Imaging. v_0 is the distance between the image plane and the lens, u_0 is the distance between the lens and the locus of perfect focus, and r is the radius of the lens. When a point at distance $u > u_0$ is projected through the lens, it focuses at a distance $v < v_0$, so that a blur circle is formed.

APPENDIX

For a thin lens,

$$\frac{1}{u} + \frac{1}{v} = \frac{1}{F} \tag{12}$$

where u is the distance between a point in the scene and the lens, v the distance between the lens and the plane on which the image is in perfect focus, and F the focal length of the lens. Thus,

$$u = \frac{Fv}{v - F} \tag{13}$$

For a particular lens, F is a constant. If we then fix the distance v between the lens and the image plane to the value $v = v_0$, we have also determined a locus of points at distance $u = u_0$ that will be in perfect focus, i.e.,

$$u_0 = \frac{Fv_0}{v_0 - F} \tag{14}$$

We may now explore what happens when a point at a distance $u > u_0$ is imaged. Figure 5 shows the situation in which a lens of radius r is used to project a point at distance u onto an image plane at distance v_0 behind the lens. Given this configuration, the point would be focused at distance v behind the lens—but in front of the image plane. Thus, a blur circle is formed on the image plane. Note that a point at distance $u < u_0$ also forms a blur circle; throughout this paper we assume that the lens system is focused on the nearest point so that u is always greater than u_0. This restriction is not necessary in the second algorithm, as overconstraint on the distance solution allows determination of whether $D = u > u_0$ or $D = u < u_0$.

From the geometry of Figure 5 we see that

$$\tan\theta = \frac{r}{v} = \frac{\sigma}{v_0 - v} \tag{15}$$

Combining Equations (13) and (15) and substituting the distance D for the variable u we obtain

$$D = \frac{Frv_0}{rv_0 - F(r + \sigma)}$$

or

$$D = \frac{Fv_0}{v_0 - F - \sigma f}$$

where f is the f-number of the lens.

The blurring of the image is better described by the point spread function than by a blur circle, although the blurring is bounded by the blur circle radius in the sense that the point spread function is less than some threshold outside of the blur circle. The point spread function is due primarily to diffraction effects, which for any particular wavelength produce wave cancellation and reinforcement resulting in intensity patterns qualitatively similar to the sinc function, $\frac{\sin r}{r}$, but with different amplitudes and periods for the "rings" around the central peak [2].

The "rings" produced by this function vary in amplitude, width and position with different states of focus and with different wavelengths. As wavelength varies these rings change position by as much as 90 degrees, so that the blue light troughs become positioned over the red light peaks, etc. Further, change in wavelength results in substantial changes in the amplitude of the various rings. Although this point spread function is quite complex, and the sum over different wavelengths even more so, our analysis shows that for white light the sum of the various functions obtained at different wavelengths has the general shape of a two-dimensional Gaussian.

Sampling effects caused by digitization are typically next in importance after the diffraction effects. The effect of sampling may be accounted for in the point spread function by convolving the above diffraction-produced point spread function with functions of the form $\frac{\sin r}{r}$. Other factors such as chromatic abberation, movement, and diffusion of photographic emulsion may also be accounted for in the final point spread function by additional convolutions.

The net effect, in light of the central limit theorem and our analysis of the sum of single-wavelength focus patterns, is almost certainly best described by a two-dimensional Gaussian $G(r, \sigma)$ with spatial constant σ. The spatial constant σ of the point spread function will be proportional to the radius of the blur circle; however, the constant of proportionality will depend on the particulars of the optics, sampling, etc. In this paper the radius of the blur circle and the spatial constant of the point spread function have been treated as identical; in practical application where recovery of absolute distance is desired the constant of proportionality k must be determined for the system and included in Equation (1) as follows:

$$D = \frac{Fv_0}{v_0 - F - \sigma k f}$$

Appendix E

Epipolar-Plane Image Analysis: A Technique for Analyzing Motion Sequences

By: Robert C. Bolles and H. Harlyn Baker

EPIPOLAR-PLANE IMAGE ANALYSIS: A TECHNIQUE FOR ANALYZING MOTION SEQUENCES*

Robert C. Bolles
H. Harlyn Baker
SRI International
333 Ravenswood Avenue
Menlo Park, CA 94025.

A technique for unifying spatial and temporal analysis of an image sequence taken by a camera moving in a straight line is presented. The technique is based on a "dense" sequence of images–images taken close enough together to form a solid block of data. Slices of this solid directly encode changes due to motion of the camera. These slices, which have one spatial dimension and one temporal dimension, have more structure in them than conventional images. This additional structure makes them easier to analyze. We present the theory behind this technique, describe an initial implementation, and discuss our preliminary results.

Introduction

Most motion-detection techniques analyze pairs of images, and hence are fundamentally similar to conventional stereo techniques (e.g., [Barnard], [Haynes], and [Hildreth]). A few researchers have considered sequences of three or more images (e.g., [Ullman], [Yen], and [Nevatia]), but still the process is one of matching discrete items at discrete times. And yet, it is widely acknowledged that there *is a potential benefit from unifying* the analysis of spatial and temporal information. In this paper we present a technique to perform this type of unification for straight-line motions.

Motion-analysis techniques using pairs or triples of images are designed to process images that contain significant changes between images - features may move 20 pixels or more from one image to the next. These large changes force the techniques to tackle the difficult problem of stereo correspondence. Our idea, on the other hand, is to take a sequence of images from positions that are very close together - close enough that almost nothing changes from one image to the next. In particular, we take images close enough together that none of the image features moves more than a pixel or so. (Figure 1 shows the first two images from one of our sequences containing 125 images.) This sampling frequency guarantees a continuity in the temporal domain that is similar to the continuity in the spatial domain. Thus, an edge of an object in one image appears temporally adjacent to (within a pixel of) its occurrence in both the preceding and following images. This temporal continuity makes it possible to construct a solid of data in which time is the third dimension and continuity is maintained over all three dimensions (see Figure 2). This solid of data is referred to as *spatio-temporal data.*

The traditional motion-analysis paradigm detects features in spatial images (i.e., the uv images in Figure 2), matches them from image to image, and then deduces the motion. We, however, propose an approach that is orthogonal to this. We suggest slicing the spatio-temporal data along a temporal dimension (see Figure 3), locating features in these slices, and then computing three-dimensional locations. Our reasoning is that the temporal image slices can be formed in such a way that they contain more structure than spatial images; thus, they are more predictable and, hence, easier to analyze.

To convince you of the utility of this approach, we must demonstrate that there is an interesting class of motions for which we can build structured temporal images. In the next section we show that this can be done whenever the camera moves in a straight line. We call these temporal images *epipolar-plane images,* or EPIs, from their geometric properties. In Section 3 we describe the results of our experiments in computing the depths of objects from their paths through EPIs. And finally, in Section 4 we discuss the strengths and weaknesses of the technique and outline some current and future directions for our work.

Epipolar-Plane Images

In this section we define an epipolar-plane image (an EPI) and explain our interest in it. First, however, we review some stereo terminology. Consider Figure 4, which is a diagram of a general stereo configuration. The two cameras are modeled as pin-holes with the image planes in front of the lenses. For each point P in the scene, there is a plane, called the *epipolar plane,* which passes through the point and the line joining the two lens centers. This plane intersects the two image planes along *epipolar lines.* All the points in the epipolar plane are projected onto one epipolar line in the first image and a corresponding epipolar line in the second image. The importance of these lines for stereo processing is that they reduce the search required to find matching points from two dimensions to one. Thus, to find a match for a point along one epipolar line in an image it is only necessary to search along the corresponding epipolar line in the other image. This is termed the *epipolar constraint.*

One further definition that is essential to understanding our approach is that of an *epipole.* An *epipole* in a stereo configuration is the intersection of the line joining the lens centers and ar image plane (see Figure 4). In motion analysis, an epipole is often referred to as a focus of expansion (FOE) because the epipolar lines radiate from it.

Consider a simple motion in which a camera moves from right to left, with its optical axis orthogonal to its direction of motion (see Figure 5). For this type of motion the epipolar plane for a point, such as P, is the same for all pairs of camera positions, and we refer to that plane as the epipolar plane for P for the whole motion.

The epipolar lines associated with one of these epipolar planes are horizontal scan lines in the images (see

*This research was supported by DARPA Contracts MDA 903-83-C-0027 and DACA 76-85-C-0004

Figure 5). The projection of P onto these epipolar lines moves to the right as the camera moves to the left. The velocity of this movement along the epipolar line is a function of P's distance from the line joining the lens centers. The closer it is, the faster it moves.

For this type of lateral motion, the epipolar lines are not only horizontal, they occur at the same vertical position in all the images. Therefore, a horizontal slice of the spatio-temporal data formed from this motion contains all the epipolar lines associated with one epipolar plane (see Figure 6).

Figure 6 shows three of the images used to form the solid of data. Typically a hundred or more images are used, making P's trajectory through the data a continuous path, as indicated in the diagram. For this type of lateral motion, if the camera moves a constant distance between images, the trajectories are straight lines (see Appendix A).

Figure 7 shows a horizontal slice through the solid of data shown in Figure 2, which was constructed from a sequence of 125 images taken by a camera moving from right to left. Figure 8 shows a frontal view of that slice. We call this type of image an epipolar-plane image (EPI) because it is composed of one-dimensional projections of the world points lying on an epipolar plane. Each horizontal line of the image is one of these projections. Thus, time progresses from bottom to top, and, as the camera moves to the left, the features move to the right.

There are several things to notice about this image. First, it contains only linear structures. In this respect it is much simpler than the spatial images used to create it (see Figure 1 for comparison). Second, the slopes of the lines determine the distances to the corresponding features in the world. The greater the slope, the farther the feature. Third, occlusion, which occurs when a closer feature moves in front of a more distant one, is immediately apparent in this representation. For example, the white bar on the right of the EPI in Figure 8 is initially occluded, then it is visible for awhile until it is occluded briefly by a thin object, then visible again before being occluded by a wide object, and at the end of the sequence is visible for a third time. Thus, the same object is seen three different times.

Figure 9 shows another EPI sliced from the data in Figure 2. Its basic structure is the same as Figure 8; however, it illustrates the variety of patterns that can occur in an EPI.

The EPIs in Figures 8 and 9 were constructed from a simple right-to-left motion with the camera oriented at right angles to the motion. For what other types of motions can EPIs be constructed? The answer is that they can be constructed for any straight-line motion. As long as the lens center of the camera moves in a straight line the epipolar planes remain fixed relative to the scene. The points in each of these planes function as a unit. They are projected onto one line in the first image, another line in the second image, and so on. The camera can even change its orientation about its lens center as it moves along the line without affecting this partitioning of the scene. Orientation changes move the epipolar lines around in the image plane, significantly complicating the construction of the EPIs, but the epipolar planes remain unchanged since the line joining the lens centers remains fixed.

Figure 10 is an EPI formed from a sequence of images taken by a camera moving forward, looking straight ahead, but down slightly. Again the image is very structured, except that, instead of lines, it is composed of curves. For this type of motion, in fact for any straight-line motion in which the camera is at a fixed orientation relative to the direction of motion (see Figure 11), the trajectories in the EPI's are hyperbolas (see Appendix B). Not only are they hyperbolas, but they are simple hyperbolas in the sense that their asymptotes are vertical and horizontal lines. A right-to-left motion, such as the one mentioned above, is just a special case in which the hyperbolas degenerate into lines.

If the lens center does not move in a line, the epipolar planes passing through a world point differ from one camera position to the next. The points in the scene are grouped one way for one pair of camera positions and a different way for another pair of positions. This makes it impossible to partition the scene into a fixed set of planes, which in turn means that it is not possible to construct EPIs for such a motion.

One last observation about EPIs: since an EPI contains all the information about the features in a slice of the world, the analysis of a scene can be partitioned into a set of analyses, one for each slice. In the case of a right-to-left motion, there is one analysis for each scanline in the image sequence. This ability to partition the analysis is one of the key properties of our motion-analysis technique. Slices of the spatio-temporal data can be analyzed independently (and possibly in parallel), and then the results can be combined into a three-dimensional representation of the scene.

Experimental Results

We have implemented a program that computes three-dimensional locations of world features by analyzing EPIs constructed from right-to-left motions. The program currently consists of the following steps:

1. 3D smoothing of the spatio-temporal data
2. Slicing the data into EPIs
3. Detecting edges, peaks, and troughs
4. Segmentating edges into linear features
5. Merging collinear features
6. Computing x-y-z coordinates
7. Building a map of free space

In this section we illustrate the behavior of this program by applying it to the data shown in Figures 1 and 2.

The first step smooths the three-dimensional data to reduce the effects of noise and camera jitter. This is done by applying a sequence of three one-dimensional Gaussians (also see [Buxton 1983] and [Buxton 1985]).

The second step forms EPIs from the spatio-temporal data. For a lateral motion this is straightforward because the EPIs are horizontal slices of the data. Figure 8 shows the EPI selected to illustrate steps three through seven.

The third step detects edge-like features in the EPI. It currently locates four types of features: positive and negative zero-crossings [Marr] and peaks and troughs in the difference of Gaussians. The zero-crossings indicate places in the EPI where there is a sharp change in image intensity, typically at surface boundaries or surface markings, and the peaks/troughs occur between these zero-crossings. The former are generally more precisely positioned than the latter. Figure 12 shows all four types of features detected in the EPI shown in Figure 8.

The fourth step fits linear segments to the edges. It does this in two passes. The first pass partitions the edges at sharp corners by analyzing curvature estimates along the edges. The second pass applies Ramer's algorithm [Ramer] to recursively partition the smooth segments into line segments. Figure 13 shows the line segments derived from the edges in Figure 12.

The fifth step builds a description of the line segments that links together those that are collinear. The intent is to identify sets of lines that belong to the same feature in the world. By bridging gaps caused by occlusions, the program can improve its estimates of the features' locations as well as extract clues about the nature of the surfaces in the scene. The program only links together features of the same type, except that positive and negative zero crossings can be joined because the contrast across an edge can differ from one view to the next. Figure 14 shows the peak features from Figure 13 that are linked together by the program.

The line intersections in Figures 13 and 14 indicate temporal occlusions. For each intersection, the feature with the smaller slope is the one that occludes the other.

The sixth step computes the x-y-z locations of the world features corresponding to the EPI features. The world coordinates are uniquely determined by the location of the epipolar plane associated with the EPI and the slope and intercept of the line in the EPI. To display these three-dimensional locations, the program can either produce a stereo pair or plot the two-dimensional coordinates of the features in the epipolar plane. Figure 15 shows the epipolar plane coordinates for the features shown in Figure 13. The shape and size of each ellipse depicts the error associated with the feature's location.

The seventh step builds a two-dimensional map of the world that indicates which areas are empty (also see [Bridwell]). The idea behind this construction is that, whenever a feature is seen by the camera, there is a clear line of sight from the camera to the feature. Therefore, if a feature is visible continuously during a portion of a motion, this line of sight sweeps out a triangle of empty space defined by the feature's location, the first position of the camera at which the feature is visible, and the last position at which the feature is visible. The program builds the map of empty space by constructing one of these triangular regions for each line segment found in an EPI, and then OR-ing them together. Figure 16 shows the map constructed for the features in Figure 15.

Discussion

The following positive characteristics of this approach should be noted:

- Spatial and temporal data are treated together as a single unit;
- The acquisition and tracking steps of the conventional motion analysis paradigm are merged into one step;
- The approach is feature-based, but is not restricted to point features - linear features that are perpendicular to the direction of motion can also be used;
- There is more structure in an EPI than in a standard spatial image, which means that it is easier to analyze, and hence easier to interpret;
- Occlusion is manifested in an EPI in a way that increases the chance of detection because the edge is viewed over time against a variety of backgrounds;
- EPIs facilitate the segmentation of a scene into opaque objects occurring at different depths because they encode a *homogeneous* slice of the object over time;
- There are some obvious ways to make the analysis incremental in time, and partitionable in y (epipolar planes), for high speed performance.

With these benefits, the inherent limitations and current restrictions must be borne in mind:

- Motion must be in a straight line and (currently) the camera must be at a fixed angle relative to the direction of motion;
- Frame rate must be high enough to limit the frame-to-frame changes to a pixel or so (more specifically, such that the projective width of the surface is greater than its motion);
- Independently moving objects will either not be detected, or will not be detected accurately.

We are currently investigating the following areas:

- Linking together features from adjacent EPIs.
- Identifying and interpreting spatial and temporal phenomena such as occlusions, shadows, mirrors, and highlights.
- Characterizing the appearance of curved surfaces in EPIs.
- Implementing the analysis of EPIs derived from forward motions.

Appendix A: Lateral-Motion Trajectories

In this appendix we first derive an equation for the trajectory of a point in an EPI constructed from a lateral motion, and then show how to compute the (x,y,z) location of such a point. Figure 17 is a diagram of a trajectory in an EPI derived from the right-to-left motion illustrated in Figure 18. The scanline at t_1 in Figure 17 corresponds to the epipolar line l_1 in Figure 18. Similarly, the scanline at t_2 corresponds to the epipolar line l_2. (Recall that the EPI is constructed by extracting one line from each image taken by the camera as it moves along the line joining c_1 and c_2. Since the images are taken very close together in time, there would be several images taken between c_1 and c_2. However, to simplify the diagram none of these is shown.) The point (u_1, t_1) in the EPI corresponds to the point (u_1, v_1) in the image taken by the camera at time t_1 and position c_1. Thus, as the camera moves from c_1 to c_2 in the time interval t_1 to t_2, the scene point *moves* in the EPI from (u_1, t_1) to (u_2, t_2). The intent of this section is to characterize the shape of this trajectory and then compute the three-dimensional position of the corresponding scene point, given the focal length of the camera, the camera speed, and the coordinates of points along the trajectory.

For our analysis we define a left-handed coordinate system that is centered on the initial position of the camera (i.e., c_1 in Figure 18). The shape of the trajectory can be derived by analyzing the geometric relationships in the epipolar plane that passes through P. Figure 19 is a diagram of that plane.

Given the speed of the camera, s, which is assumed to be constant, the distance from c_1 to c_2, Δx, can be computed as follows:

$$\Delta x = s\,\Delta t \qquad (1)$$

where Δt is $(t_2 - t_1)$. By similar triangles

$$\frac{u_1}{h} = \frac{x}{D} \quad (2)$$

$$\frac{u_2}{h} = \frac{\Delta x + x}{D} \quad (3)$$

where u_1 and u_2 have been converted from pixel values into distances on the image plane, h is the distance from the lens center to the epipolar line in the image plane, x is the x-coordinate of P in the scene coordinate system, and D is the distance from P to the line joining the lens centers. Since h is the hypotenuse of a right triangle, it can be computed as follows:

$$h = \sqrt{f^2 + v_1^2} \quad (4)$$

where f is the focal length of the camera. From 2 and 3 we get

$$\Delta u = (u_2 - u_1) = \frac{h(\Delta x + x)}{D} - \frac{h\,x}{D} = \frac{h}{D}\Delta x \quad (5)$$

Thus, Δu is a linear function of Δx. Since Δt is also a linear function of Δx, Δt is linearly related to Δu, which means that trajectories in an EPI derived from a lateral motion are straight lines.

The (x, y, z) position of P can be computed by scaling u_1, v_1, and f appropriately. From 5 we define

$$m = \frac{D}{h} = \frac{\Delta x}{\Delta u} \quad (6)$$

which represents the slope of the trajectory computed in terms of the distance traveled by the camera (Δx as opposed to Δt) and the distance the point moved along the epipolar line (i.e., Δu). From similar triangles

$$(x, y, z) = \left(\frac{D}{h}u_1, \frac{D}{h}v_1, \frac{D}{h}f\right) \quad (7)$$

which means that

$$(x, y, z) = (mu_1, mv_1, mf) \quad (8)$$

If the first camera position, c_1, on an observed trajectory is different from the camera position, c_0, that defines a global camera coordinate system, the x coordinate can be adjusted by an amount equal to the distance traveled from c_0 to c_1. Thus,

$$(x, y, z) = ((t_1 - t_0)s + mu_1, mv_1, mf) \quad (9)$$

where t_0 is the time of the first image and s is the speed of the camera. This correction is equivalent to computing the x intercept of the line and using it as the first camera position. Therefore, for a lateral motion, the trajectories are linear and the (x, y, z) coordinates of the points can be easily computed from the slopes and intercepts of the lines.

Appendix B: Forward-Motion Trajectories

The derivation of the form of a trajectory produced by a forward motion is similar to the one used for lateral motion. Figure 20 is a diagram of a trajectory in an EPI derived from a sequence of images taken by a camera moving in a straight line at a fixed orientation relative to the principal axis of the camera (see Figure 21). Without loss of generality we have rotated the image plane coordinate systems in a uniform way so that the epipoles are on the u axes. The EPI in Figure 20 was constructed by extracting pixel intensities along epipolar lines in the images shown in Figure 21 and inserting them as scanlines in Figure 20. For example, epipolar line l_1 was placed at t_1, l_2 was placed at t_2, and so on. The point (w_1, t_1) in the EPI corresponds to the point (u_1, v_1) in the image taken at time t_1 and position c_1. Thus, as the camera moves from c_1 to c_2 over the time interval t_1 to t_2, the scene point *moves* in the EPI from (w_1, t_1) to (w_2, t_2). Our goal is to characterize the shape of this trajectory, and then compute the three-dimensional position of the corresponding scene point, given the focal length of the camera, the camera speed, the angle between the camera's axis and the direction of motion (θ), and the coordinates of points along the trajectory.

As before, we define a left-handed coordinate system that is centered on the initial position of the camera (i.e., c_1 in Figure 21). The shape of the trajectory can be derived by examining the geometric relationships in the epipolar plane that passes through P. Figure 22 is a diagram of that plane.

Given the speed of the camera, s, which is assumed to be constant, the distance from c_1 to c_2, Δe, can be computed as follows:

$$\Delta e = s\,\Delta t \quad (10)$$

where Δt is $(t_2 - t_1)$. By similar triangles

$$\frac{w_1}{h} = \frac{C}{e} \quad (11)$$

and

$$\frac{w_2}{h} = \frac{C}{(e - \Delta e)} \quad (12)$$

where w_1 and w_2 are distances on the image plane, h is the distance from the lens center to the epipole, C is the distance from P to the line joining the lens centers measured in a plane parallel to the image planes, and e is the distance along the line joining the lens centers from c_1 to the plane passing through P and parallel to the image planes. Since h is the hypotenuse of a right triangle (see Figure 21), it can be computed as follows:

$$h = \frac{f}{cos\theta} \quad (13)$$

where f is the focal length of the camera. From 11 and 12 we get

$$\Delta w = (w_2 - w_1) = \frac{hC}{(e - \Delta e)} - \frac{hC}{e} = \frac{hC\Delta e}{e(e - \Delta e)} \quad (14)$$

which can be rewritten as

$$e\,\Delta w\,\Delta e \;-\; e^2\Delta w \;+\; hC\Delta e \;=\; 0 \quad (15)$$

Using 10 to express Δe in terms of Δt, this becomes

$$se\Delta w\Delta t \;-\; e^2\Delta w \;+\; shC\Delta t \;=\; 0 \quad (16)$$

which defines a hyperbola whose asymptotes are the lines $w = 0$ and $t = e/s$ (see Figure 23). Thus, the trajectory is a hyperbola in which the point P appears arbitrarily close to the epipole when the camera is far away from it (as one would expect), and the projection of P moves away from the epipole at an increasing rate as the camera gets closer to it. This relationship agrees intuitively with the fact that a projective transformation involves a $1/z$ factor, which makes u a hyperbolic function of z.

Equation 14 can be used to compute z. First, rewrite it as follows:

$$\Delta w = \frac{hC}{(e - \Delta e)}\frac{1}{e}\,\Delta e \quad (17)$$

Then using Equation 12 and

$$e = \frac{z}{cos\theta} \tag{18}$$

we get

$$\Delta w = \frac{w_2\, cos\theta\, \Delta e}{z} \tag{19}$$

or

$$z = \frac{w_2\, cos\theta\, s \Delta t}{\Delta w} \tag{20}$$

Notice that it is NOT necessary to determine the coefficients of the hyperbola in order to compute z. Two points on the trajectory are sufficient to compute Δt and Δw, which in turn, are sufficient to compute z. Also notice, however, that it is easy to fit an hyperbola of this type because it is in the simple form

$$\Delta w \Delta t + a \Delta w + b \Delta t = 0 \tag{21}$$

which is linear with respect to the coefficients a and b. This type of fitting provides a way to increase the precision with which the scene points are located.

The expression for z in Equation 20 does not apply when $\theta = 90°$, but that is the lateral motion case covered earlier. Thus, the trajectories are always hyperbolas; they just happen to degenerate into straight lines when $\theta = 90°$, which corresponds to the case in which the epipoles are not in the image plane, but rather lie at infinity.

The x and y coordinates for P can be computed by scaling z appropriately:

$$(x, y) = (\frac{u_1}{f} z, \frac{v_1}{f} z) \tag{22}$$

Recall that u_1 and v_1 are measured in a rotated-image-plane coordinate system that was set up to place the epipole on the u axis. Therefore, in addition to converting pixel values to a standard metric, such as meters, the image coordinates of a point must be rotated about the principal axis before they can be inserted into Equation 22. To compute a world-centered position for P, the (x, y, z) position computed by Equations 20 and 22 has to be transformed for the initial position of the camera along the path.

References

[Barnard] "Disparity Analysis of Images," S. T. Barnard and W. B. Thompson, *IEEE Trans., PAMI,* Vol 2, No 4, 1980.

[Bridwell] "A Discrete Spatial Representation for Lateral Motion Stereo," N. J. Bridwell and T. S. Huang, *Computer Vision, Graphics, and Image Processing,* Vol 21, 1983.

[Buxton 1983] "Monocular Depth Perception from Optical Flow by Space Time Signal Processing," B. F. and Hilary Buxton, *Proceedings of the Royal Society of London, B.,* Vol 218, 1983.

[Buxton 1985] "Computation of optic flow from the motion of edge features in image sequences," B. F. and H. Buxton, *Image and Vision Computing,* Vol 2, No 2, May 1985.

[Haynes] "Detection of Moving Edges," S. M. Haynes and R. Jain, *Computer Vision, Graphics, and Image Processing,* Vol 21, No 3, 1983.

[Hildreth] "Computations Underlying the Measurement of Visual Motion," E. C. Hildreth, *Artificial Intelligence,* Vol 23, 1984.

[Marr] "Theory of Edge Detection," D. C. Marr and E. Hildreth, *Proceedings of the Royal Society of London,* B 207, 1980.

[Nevatia] "Depth Measurement from Motion Stereo," Ramakant Nevatia, *Computer Graphics and Image Processing,* 5, 1976.

[Ramer] "An Iterative Procedure for the Polygonal Approximation of Plane Curves," U. Ramer, *Computer Graphics and Image Processing,* Vol 1, 1972.

[Ullman] *The Interpretation of Visual Motion,* S. Ullman, MIT Press, Cambridge, Mass., 1979.

[Yen] "Determining 3-D Motion and Structure of a Rigid Body Using the Spherical Projection," B. L. Yen and T. S. Huang, *Computer Vision, Graphics, and Image Processing,* Vol 21, 1983.

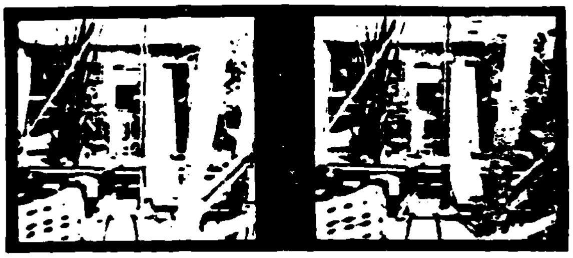


Fig. 1. First two images in a sequence

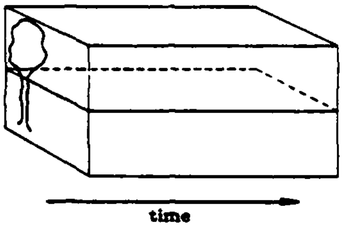


Fig. 3. Slice of the solid of data

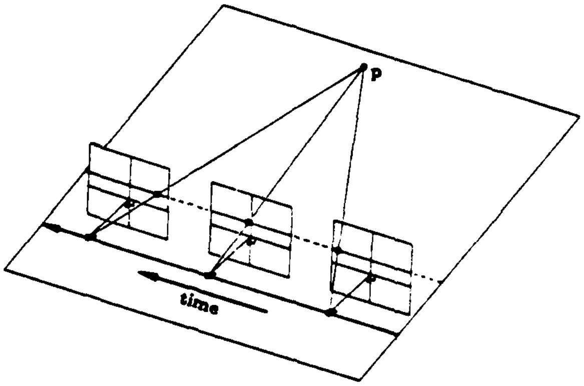


Fig. 5. Right-to-left motion

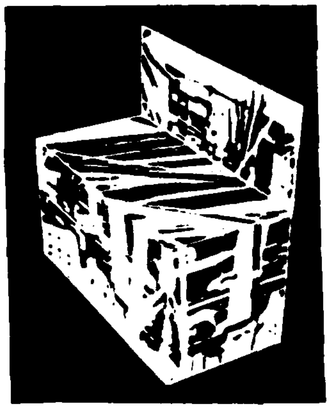


Fig. 7. Sliced solid of data

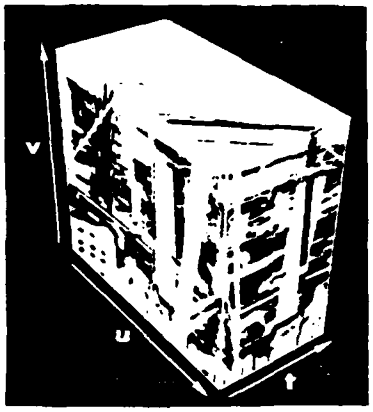


Fig. 2. Spatio-temporal solid of data

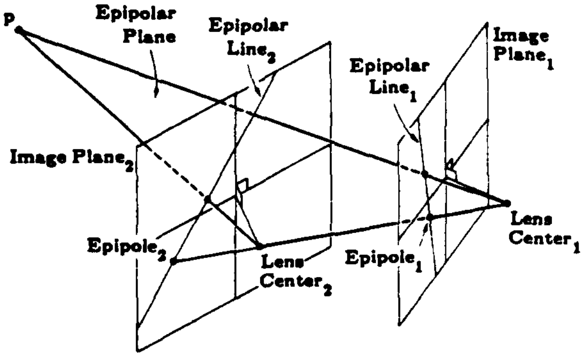


Fig. 4. General stereo configuration

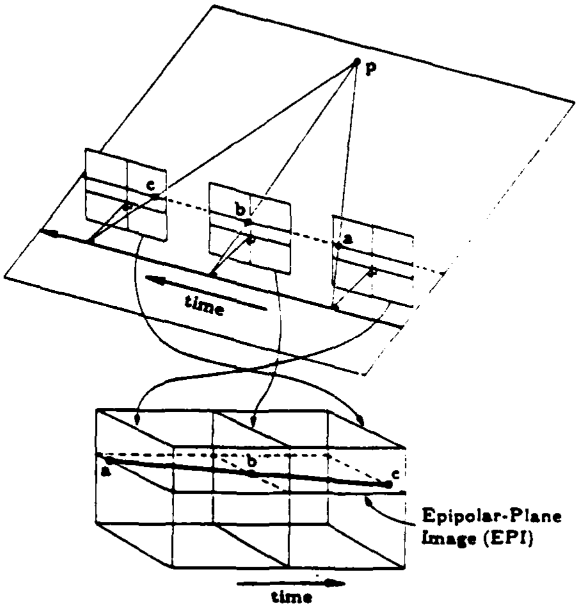


Fig. 6. Right-to-left motion with solid

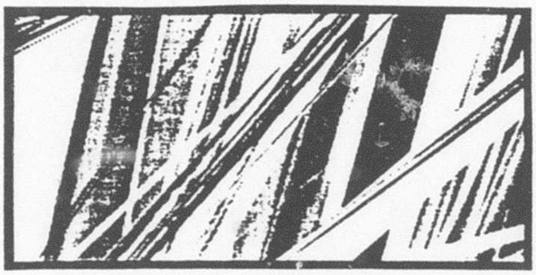


Fig. 8. Frontal view of the EPI

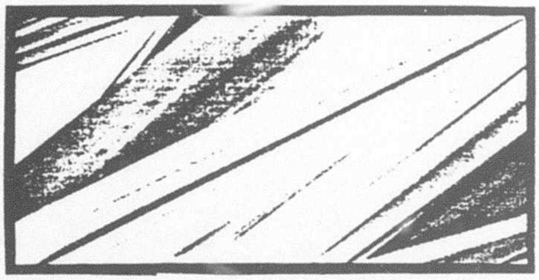


Fig. 9. A second EPI

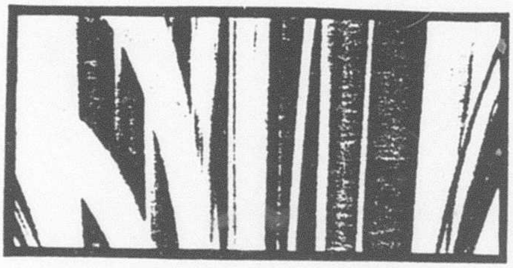


Fig. 10. EPI from forward motion

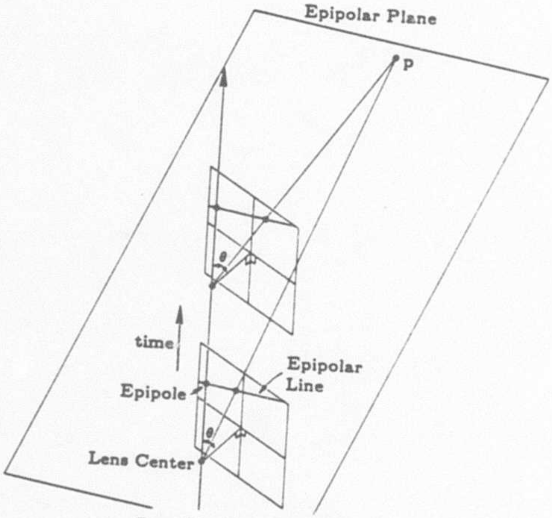


Fig. 11. Forward motion

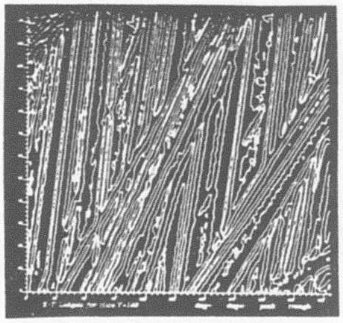


Fig. 12. Edge features in EPI

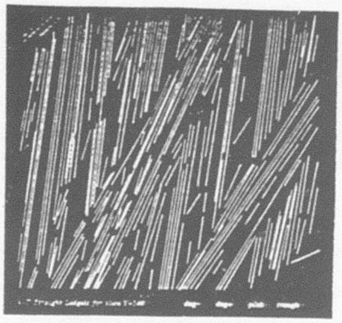


Fig. 13. Straight lines

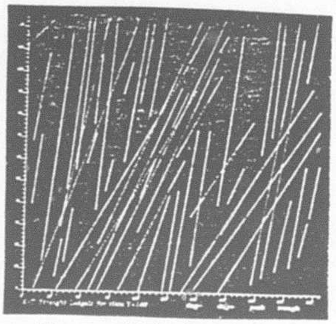


Fig. 14. Linked peak lines

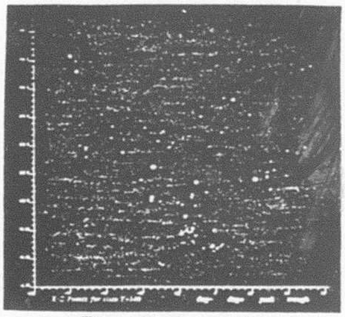


Fig. 15. xz locations

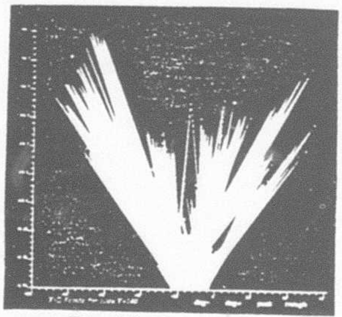


Fig. 16. Free space

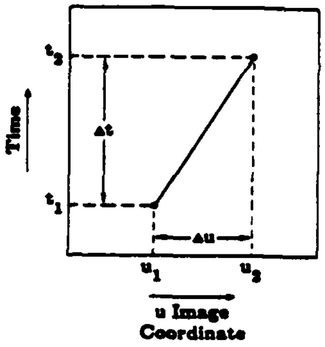


Fig. 17. Lateral motion EPI

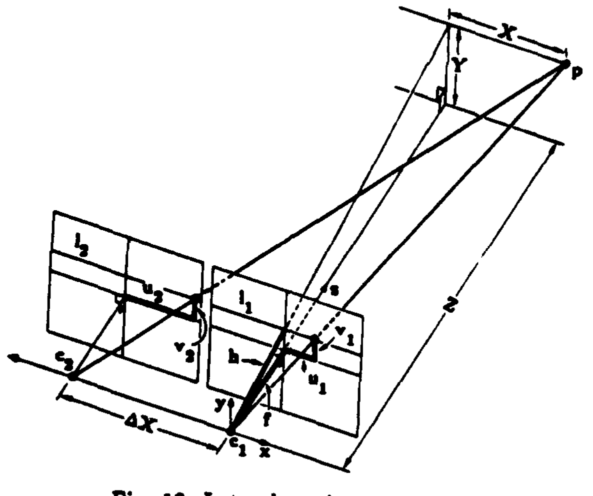


Fig. 18. Lateral motion geometry

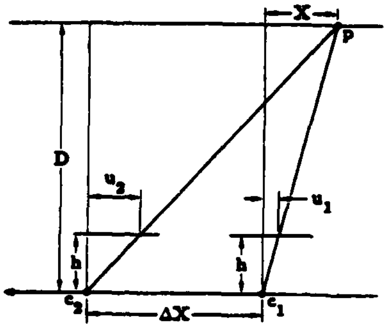


Fig. 19. Lateral motion epipolar plane

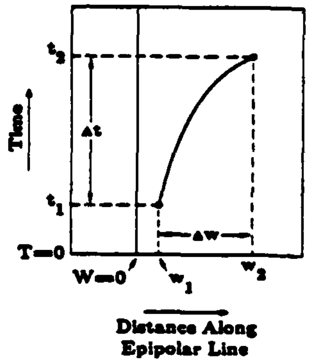


Fig. 20. Forward motion EPI

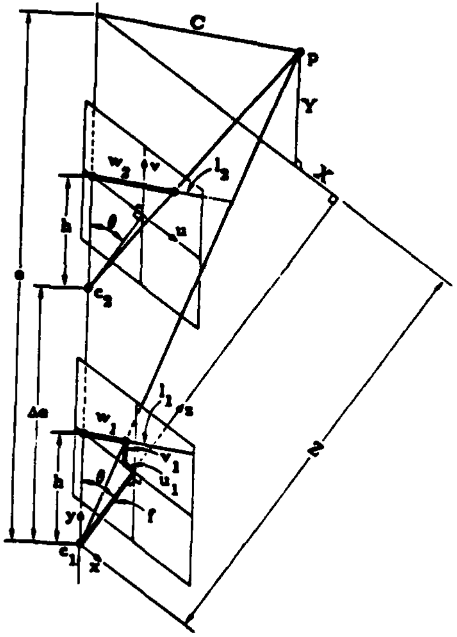


Fig. 21. Forward motion geometry

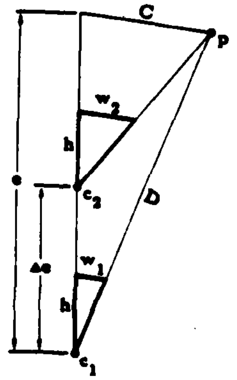


Fig. 22. Forward motion epipolar plane

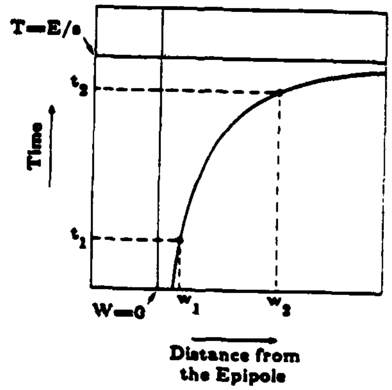


Fig. 23. Asymptotes for the hyperbola

Appendix F

An Inductive Approach to Figural Perception

By: Stephen T. Barnard

AN INDUCTIVE APPROACH TO FIGURAL PERCEPTION

Technical Note No. 325

September 19, 1984

BY: Stephen T. Barnard, Senior Computer Scientist

Artificial Intelligence Center
Computer Science and Technology Division

SRI Project 5355

The work reported herein was supported by the Defense Advanced Research Projects Agency under Contract No. MDA903-83-C-0027.

Abstract

The problem of interpreting single images of abstract figures is addressed. It is argued that neither rule-based deductive inference nor model-based matching are satisfactory computational paradigms for this problem. As an alternative, an inductive approach consisting of two parts is presented. The first part involves a scheme, based on differential geometry, for describing the shapes of curves and surfaces, and for generating these descriptions from images. The second part of the approach relies on a criterion for deciding which description, among the candidates allowed by the constraints in the image, is to be preferred. This criterion — minimum entropy — is related to concepts from Gestalt psychology, thermodynamics, and information theory. Several examples are given to illustrate the inductive approach.

Contents

List of Figures

1. Introduction

Images arise when light that encodes structure in the three-dimensional world is projected onto a photosensitive surface. Some of the information in the light is lost, and the remainder is transformed by perspective into a pattern that has a complex and ambiguous formal relationship to the original structure of the world. The human visual system is capable of inverting this relationship, filling in parts that are missing, arranging parts that are seen into sensible combinations, and, in short, composing integrated, consistent descriptions of the world, which are almost never in serious error. Furthermore, these descriptions specify invariant properties of the scene that are independent of the observer (size, shape, etc.), while the information used to construct the descriptions — the image — is highly dependent on the observer's position, orientation, and imaging system.

How is this possible? What kinds of computational strategies, representations, and modes of reasoning are appropriate to this problem, and how can they be implemented and demonstrated on a large class of examples, including images of natural scenes?

Rule-based deductive reasoning — the conventional AI paradigm — does not appear to be a good approach to perception. Because an image does not logically entail any particular interpretation, one cannot cast the problem of perception in a simple deductive model: interpretations are neither true nor false; they are only likely in varying degrees. But our perception at any moment is unambiguous. Furthermore, our perception sometimes jumps to unwarranted conclusions, as we know from many illusions. [1]

The logical basis of perception is induction. As a mode of reasoning, induction is completely different from deduction. While deduction proceeds from the general (axioms) to the particular (propositions), induction proceeds from the particular to the general. Deduction is primarily a matter of proving theorems, while induction is one of recognizing patterns. Deduction is well-understood and more easily automated with computers, which probably explains its popularity in AI research. The mathematical foundations of induction, by contrast, are much less clear. Nevertheless, general principles of inductive reasoning do exist.

It has been postulated that the uniformity and regularity of the world are necessary presuppositions of induction. This is precisely the state of affairs in perception. The underlying reality (the scene) is not logically deducible from the image, but, in most cases, a very good guess can be made by finding the simplest possible interpretation.

Specifically, The problem of **figural perception** is defined as deciding how to assign three-dimensional properties — size, shape, position, orientation, etc. — to initially two-dimensional patterns of data. The patterns of interest vary in their degree of complexity. For example, they might be simply binary contours, such as Figure 1. The sense of realism in even these simple figures compels one to believe that very general perceptual processes apply. A somewhat more complex class of patterns is synthetic intensity images, such as Figure 2, in which a combination of surface, lighting, and projection models produces images that evoke an even more vivid impression of three-dimensional shape.

Figures 1 and 2 are synthetic: they were generated with the techniques of computer graphics [1]. The Bumpy Torus, for example, was created by constructing a smooth, randomized toroidal surface, defining a reflectance function with lambertian and specular com-

[1] In a strictly logical sense, perception *always* jumps to unwarranted conclusions.

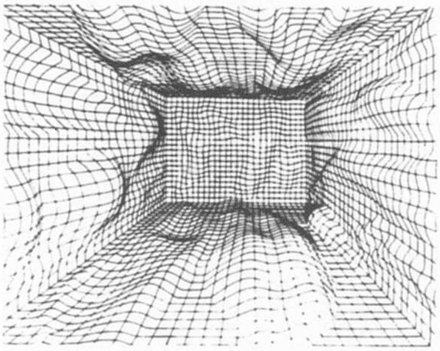


Figure 1: Wire Room

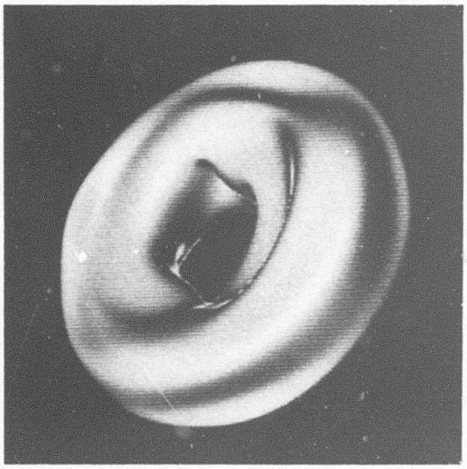


Figure 2: Bumpy Torus

ponents, defining a lighting model and a viewing position, and, finally, centrally projecting the intensities of a very fine mesh of surface points onto a synthetic digital image. A depth buffer was used to handle hidden surface areas. Using synthetic data has two important methodological advantages: (1) the underlying reality is known to arbitrary precision and can easily be used to evaluate interpretations, and (2) variables that are difficult to control in physical imaging, such as lighting and film response, are easily controlled in a synthetic regime. Of course, if a theory of figural interpretation is to have practical importance, it must be applicable to real images. If a computational vision technique works well on very realistic synthetic images, without relying on special conditions that are known *a priori* (such as a specific lighting model), then it will probably work well on comparable real images. If the technique shows improved performance on images that are subjectively more realistic, we can be even more confident that it will be valid for real images.

The physical constraints in the problem of figural perception, while obviously important, are insufficient: infinitely many possible surfaces could have caused these figures, but our perception chooses only one. The thesis behind this paper is that a formal geometrical language, together with general principles of inductive reasoning, can account for at least a large part of the solution to this underdetermined problem. A geometrical language, combined with physical constraints, provides a space of possible three-dimensional descriptions or "explanations" of patterns, and inductive reasoning provides a basis for choosing among them.

The inductive approach to figural perception has two critical elements:

- First, there is a representational scheme, based on vector algebra and differential geometry, that can model the image and all of its possible interpretations. Implicit in this scheme is a process for *generating* interpretations (Section 3).

- Secondly, there is an inductive criterion for preferring certain interpretations over others. This criterion — minimum entropy — is based on a formalism originally developed for statistical mechanics. In the context of figural perception, entropy is used as a measure of disorder (Section 4).[2]

The approach treats perception as a search for the simplest explanation of a body of data (an image). An interpretation is therefore a re-encoding of an image. Properties and relations that are explicit or easily computed from the image (pixel values, edges, textural properties, etc.) become implicit in the re-encoding and may be at least partially recovered by reprojection. On the other hand, properties and relations that are merely implicit in the image (scene invariants, such as shape, size, orientation, relative position, reflectivity, transparency, etc.) are explicit in the re-encoding. The image is unstructured and lengthy: it contains redundant information. The re-encoding is structured and terse: it contains at least as much information as the image, and usually more, but in a compressed form, with the redundant part removed. Some process not yet fully understood *discovers* redundancy in the image and *exploits* this redundancy to build more concise and well-formed encodings. In practice, it may not be necessary to actually construct a concise encoding, but merely to recognize that one is possible.

It is useful to think of an agent that "decodes" the final interpretation and that has the knowledge and ability of a computer graphics system. The 3D encoding describes the scene in terms of physically meaningful, invariant properties. The agent can decode it, in principle, into a "visualization" of the scene by using an abstract model of projection, a choice of viewpoint and lighting, and specific knowledge of physical principles, such as that an opaque object occludes what is behind it or that a transparent object transmits light. Therefore, while the interpretation contains no less information than the image, it is in a form that makes the important invariant properties explicit, and relegates the ones that depend on viewpoint and lighting to an implicit status.

2. Related Work

Two distinctly different schools of research have addressed the problem of figural perception. The artificial intelligence (AI) approach has focused on computer implementations, while the perceptual psychology approach has developed primarily theoretical models. The scientific methods used in the two disciplines are quite different. Vision research in the AI style generally requires precise computational models of perception: if a theory cannot be implemented, it is too vague to be of value. Ultimately, the model should be evaluated on images of real scenes. Vision research in perceptual psychology, by contrast, has sought to explain *human* perception as revealed by illusions, psychophysical experiments, and introspection. Perceptual psychology is by far the older school, and AI has borrowed from it liberally. At the same time, the development of computers has influenced psychologists to pursue information-processing approaches and to embrace concepts originally developed in AI [5].

2.1. AI

The deductive approach to figural perception has been explored in the so-called "blocks world" work (see Mackworth [2] for a summary of this research), culminating in Waltz's fil-

[2] While the representational scheme is based on the geometry of curves and surfaces, the reasoning scheme has far broader generality.

tering technique for constraint satisfaction [3], and Kanade's generalization to the Origami world [4]. The results are not encouraging. In addition to the problem of needing a perfect line drawing to begin with, these systems produced only weak interpretations, not including, for example, quantitative estimates of length and orientation. When generalized only slightly, Waltz's filtering scheme led to many more ambiguous interpretations.

Another line of AI research, which is more relevant to the approach described here, has sought *metric* interpretations of images, as opposed to the weaker, merely descriptive interpretations characteristic of the blocks world. The first instance of such an approach was due to Huffman [6], who suggested the concept of **dual space**, later generalized by Mackworth [7] to **gradient space**. Gradient space simply provides a way of representing with two parameters the orientations of planes. Mackworth connected observed features in image space (vertices) with contraints in gradient space (i.e., constraints on the orientations of planes) to disambiguate blocks-world interpretations. Kanade used gradient space to estimate orientations on the basis of symmetry [8]. That is, image figures that exhibit skewed symmetry (because of the distortion introduced by projection) are interpreted as being oriented in a way that is consistent with their true symmetry.

This general approach — identifying an important invariant property in the plane, backprojecting image features to planes of different orientations, and selecting the orientation leading to the most well-formed configuration — has been followed by several researchers. Kender [9] used textural properties, such as the lengths and orientations of line segments; Ikeuchi [10] and Barnard [11] used angles; Witkin [12] sought the planar orientation that had the most uniform distribution of directions of contour tangents; Brady and Yuille [13] maximized the compactness of the backprojected closed contour; and Barnard [11] maximized the uniformity of backprojected curvature. The inductive approach can possibly unify these various criteria into a single principle.

Another area of AI vision research that is relevant to figural perception is the optimal interpolation of surfaces [14], [15],[16], [17]. The mathematical representation of surfaces and the optimization methods used in this work have similarities to the approach described here. The underlying problems are quite different, however. The problem of optimal interpolation is to begin with sparse three-dimensional data (distances and orientations), presumably derived from stereo, shape-from-shading analysis, etc., and to find a continuous surface that best fits the data, while optimizing physical properties of the surface (specifically, potential energy). The problem of figural perception initially provides no three-dimensional information at all, and is not even well-posed in the sense that the interpolation problem is. Furthermore, we choose interpretations according to their simplicity of description, and not according to a physical property.

2.2. Perceptual Psychology

A popular approach in perceptual psychology has sought to exploit the efficacy of information theory [18], [19], [20], [21], [22], [23], [24]. Rock calls this the modern version of Gestalt theory ([5], p. 133), because its aim, just like Gestalt, is to explain perception in terms of simplicity. While there is not space here to cover all this work, it will be useful to discuss in some detail a recent approach that has some similarities to the approach presented here.

Buffart, et. al. presented a "coding theory" of perception that was meant to explain

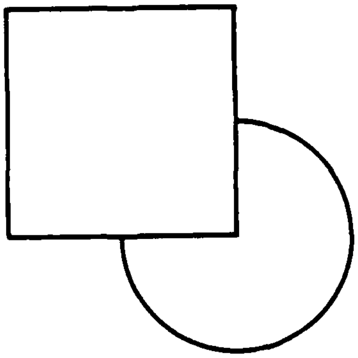


Figure 3: The Interposition Illusion

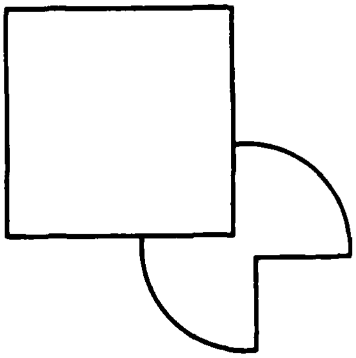


Figure 4: Kanizsa's Counter Example

the interposition illusion [25]. Most observers see the pattern in Figure 3 as a square on top of a circle. Coding theory attempts to explain this by asserting that a description in terms of a square on top of a circle is simpler than any other description that accounts for the figure. The authors proceed to develop a coding scheme for these figures that takes advantage of symmetries and that leads to very concise encodings. The encodings are sentences in a formal language, with the primitives representing sides, angles, circular arcs, and combinational operators. Some context-sensitive elements are included; for example, a side can be extended indefinitely until it encounters another contour. The goodness of an encoding is determined by simply counting the number of symbols it uses.

There are several objections to this theory. First, Kanizsa [26] argues that a pattern such as Figure 4 is a counter example, because the interpretation without interposition is simpler than the one with interposition: the circle with two "bites" taken from it has two axes of symmetry, and should, therefore, be more symmetric, and hence simpler, than the one with only one bite. As will be shown in Section 4.2, this objection is not valid. That a figure has more axes of symmetry than another does not imply it is simpler.

A second, more serious objection to the coding theory is that it depends on an *ad hoc* language, and there is no compelling reason to adopt this language in preference to any other. A third objection is that, even given this particular language, mere symbol counting is not a good way to measure the complexity of an encoding. A fourth objection is that no procedure for actually constructing a minimal encoding is presented. The approach presented below, when considered as an alternative to the coding theory, meets these objections.

3. A Representational Scheme for Figural Perception

The view of perception as a computational process of building, testing, and selecting descriptions is arguably the most important contribution of artifical intelligence to perceptual psychology. When faced with the task of actually implementing a computational model of perception, one must deal with representational problems that are otherwise too easily ignored. If perception is description building, what must these descriptions be like? In

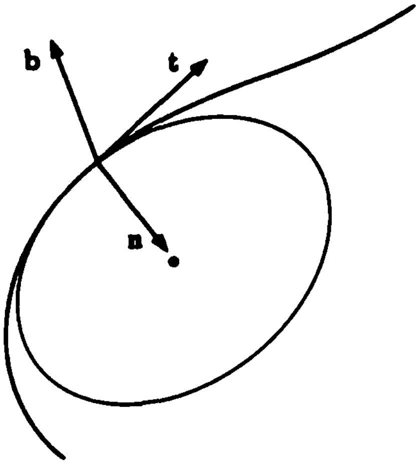


Figure 5: The Moving Trihedron

what kind of language should they be expressed?

3.1. Geometrical Descriptions

The problem of figural perception, is to a large extent, a problem of geometrical description. We seek interpretations in terms of geometrical objects: points, curves, and surfaces. The description of the special cases of points, straight lines, and planes is relatively straightforward: these objects can be represented with vector algebra [27]. Much more difficult is the representation of general curves and surfaces.

Differential geometry is the study of geometric figures using the methods of calculus [28]. Three requirements compel us to use the language of differential geometry in our representational scheme:

- If we are to compare descriptions on the basis of simplicity, we must have canonical descriptions. The descriptions must be ***unique.***

- The language must be expressive enough to describe the entire range of figural phenomena. It must be ***complete.***

- The descriptions should express ***intuitive*** and ***invariant*** figural properties.

The form of the invariant properties of curves and surfaces embedded in three-dimensional Euclidean space is completely known for our purposes.

Any curve $\mathbf{x}(s)$ in C^2 (i.e., any twice-differentiable curve) can be represented with two invariant local properties, curvature κ and torsion τ, that are scalar functions of arc length, s, and that constitute a complete, unique, and invariant representation of the curve. The relationships are described by the Serret-Frenet equations:

$$\begin{aligned} \dot{\mathbf{t}} &= \kappa \mathbf{n} \\ \dot{\mathbf{n}} &= -\kappa \mathbf{t} + \tau \mathbf{b} \\ \dot{\mathbf{b}} &= -\tau \mathbf{n} \end{aligned} \quad (1)$$

where **t**, **n**, and **b** are, respectively, the tangent, normal, and bi-normal vectors (Figure 5). The dot operator indicates differentiation with respect to arc length. The important point is that a description of a curve in terms of curvature and torsion is independent of the choice of a coordinate system. Barnard and Pentland [29] have studied the interpretation of images of 3D curves with torsion by using local assumptions of maximally uniform curvature and constant torsion.

Using the concepts of differential geometry, a surface $\mathbf{x}(u,v)$ in C^2 can also be represented with invariant local properties. The relationships analogous to the Serret-Frenet equations are the Gauss-Weingarten equations:

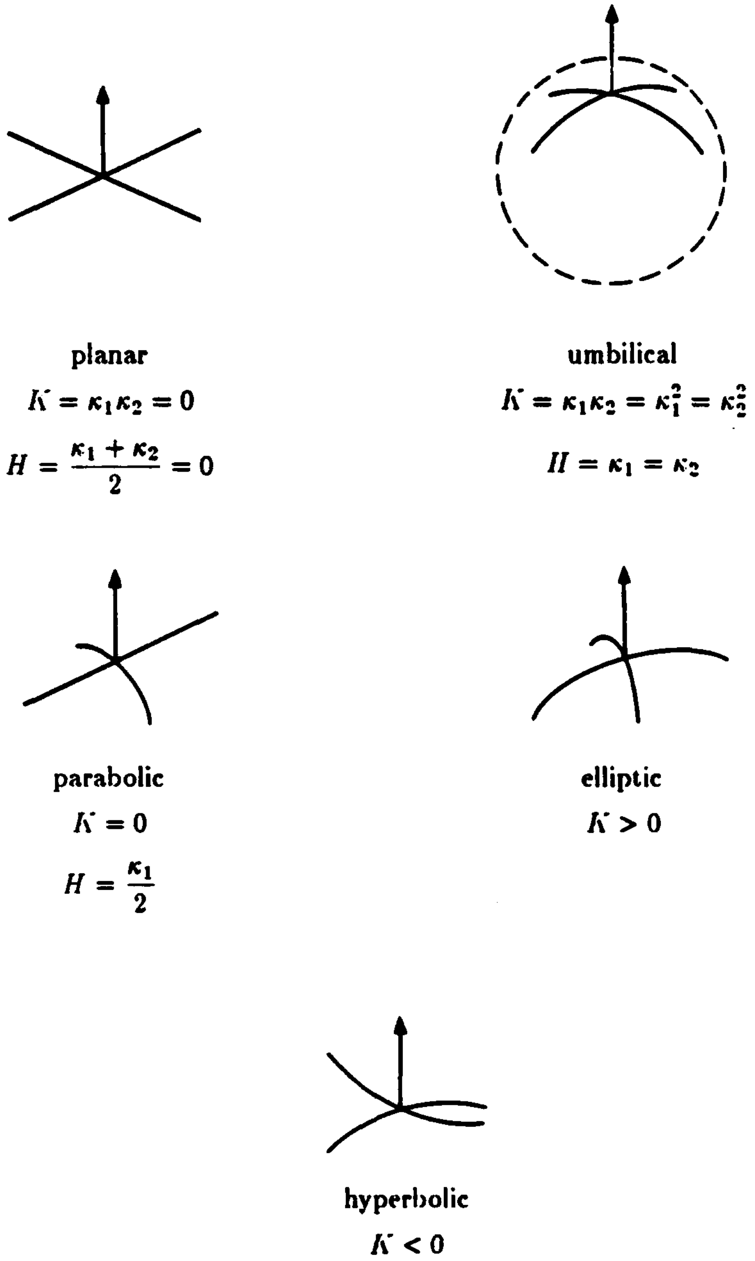
$$\begin{aligned}
\mathbf{x}_{uu} &= \Gamma^1_{11}\,\mathbf{x}_u + \Gamma^2_{11}\,\mathbf{x}_v + L\mathbf{N} \\
\mathbf{x}_{uv} &= \Gamma^1_{12}\,\mathbf{x}_u + \Gamma^2_{12}\,\mathbf{x}_v + M\mathbf{N} \\
\mathbf{x}_{vv} &= \Gamma^1_{22}\,\mathbf{x}_u + \Gamma^2_{22}\,\mathbf{x}_v + N\mathbf{N} \\
\mathbf{N}_u &= \beta^1_1\,\mathbf{x}_u + \beta^2_1\,\mathbf{x}_v \\
\mathbf{N}_v &= \beta^1_2\,\mathbf{x}_u + \beta^2_2\,\mathbf{x}_v
\end{aligned} \qquad (2)$$

where **N** is the unit normal to the surface, and the subscripts u and v indicate partial differentiation. The coefficients Γ^k_{ij}, β^j_i, L, M, and N are determined by the local shape of the surface. The theory of surfaces is much more elaborate than the theory of curves, as a comparison of Equations (1) and (2) suggests.

To develop an intuitive understanding of the power of the theory, consider the concepts of normal curvature, geodesic curvature, principal curvature, gaussian curvature, and mean curvature. The unit normal to a surface, **N**, at a point P, defines a plane tangent to the surface at P. Any line through P in this plane locally determines a curve on the surface, and hence a **normal curvature** κ_n. The normal curvature will be a maximum in one direction and a minimum in the orthogonal direction.[3] These are called the **principal directions**, and the corresponding normal curvatures κ_1 and κ_2, the **principal curvatures**. The quantity $K = \kappa_1\kappa_2$ is called the **gaussian curvature**, and the quantity $H = \frac{1}{2}(\kappa_1 + \kappa_2)$ is called the **mean curvature**. Figure 6 illustrates the connection between gaussian and mean curvature and intuitive ideas about the qualitative shapes of surfaces. A curve through P that connects two points Q and R by the shortest path is called a **geodesic**, and, when it is orthogonally projected onto the tangent plane at P, it forms (locally) a straight line, or, equivalently, a curve of zero curvature. If *any* curve on the surface through P is projected onto the tangent plane, the curvature of the resulting planar curve is called the **geodesic curvature**. Geodesic curvature and gaussian curvature are intrinsic properties of surfaces.

The qualitative shape of surfaces is suggested by local contours, but the precise shape is very ambiguous. Perception of figures like the Wire Room (Figure 1) seems to depend on global judgments. Perception of particular elements of the figure is preceded by, or depends upon, perception of the figure as a whole what the Gestalt psychologists called *Prägnanz*. It is possible to obtain, for example, estimates of surface normals using local information [30]. If the "goodness" of the resulting surface description can be estimated, it should be possible to find a global optimum by variational methods (for example, iterative improvement methods such as steepest descent, or more sophisticated optimization methods such as simulated annealing [31]).

[3] This is not strictly true. The surface may be planar or umbilical at P, in which case κ_n is uniform.



planar

$K = \kappa_1 \kappa_2 = 0$

$H = \frac{\kappa_1 + \kappa_2}{2} = 0$

umbilical

$K = \kappa_1 \kappa_2 = \kappa_1^2 = \kappa_2^2$

$H = \kappa_1 = \kappa_2$

parabolic

$K = 0$

$H = \frac{\kappa_1}{2}$

elliptic

$K > 0$

hyperbolic

$K < 0$

Figure 6: Local Surface Types

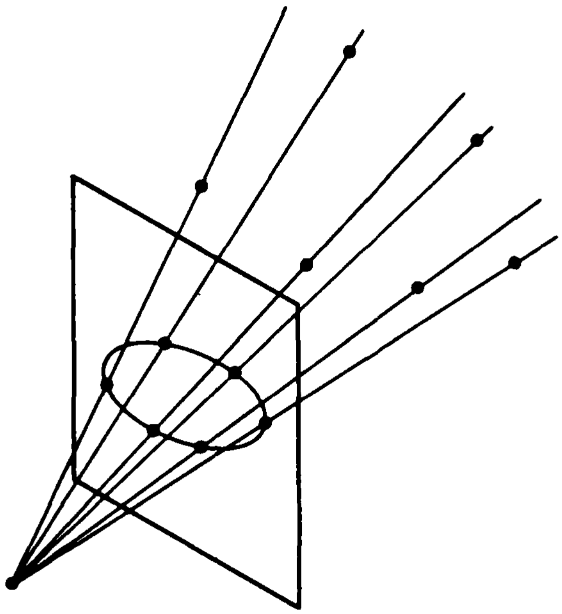


Figure 7: Wire-Bead Backprojection

3.2. Generating Hypothetical Descriptions

Even the simplest image represents an infinity of possible 3D scenes. If continuous scene space is quantized appropriately, the discrete space of possible scenes is infinite but denumerable. The class of methods for generating descriptions of these possibilities is **backprojection**. In general, any method that generates three-dimensional descriptions (in terms of distances, orientations, lighting models, reflectance models, etc.) while maintaining consistency with the geometrical and physical constraints of the image, is an instance of backprojection.

Perhaps the easiest way to visualize backprojection is with the "wire-bead" model [32] (Figure 7). Points on the image contour can be backprojected, or placed in 3D space, anywhere along a line connecting the center of projection and the image point. The wire-bead model maintains the most primitive projective constraints, but does not, for example, require connected image contours to backproject to connected 3D contours. A problem with the wire-bead model is that it allows too many degrees of freedom: one for every contour point.

Another form of backprojection is aimed at generating 3D descriptions in terms of different planar orientations (Figure 8). Assuming the image contour is the projection of a more-or-less planar contour in the scene, which is at some indeterminate distance from the observer, planar backprojection generates scale-invariant descriptions of the possible 3D contours. In the simplest case such a system has two degrees of freedom: the coordinates of the unit normal vectors of the planes. Furthermore, if the parameter space is represented as the gaussian sphere (as opposed to gradient space), the space of possibilities is closed — an important property when sampling the space at a finite number of points [11].

Another form of backprojection has been used to find the most orthogonal interpretation of image line segments (see Barnard, [33]). If linear image features can be interpreted as projections of mutually orthogonal lines in 3D space, human observers have a strong tendency to interpret them in this way [34], [35]. The effect is clearly demonstrated in the familiar Ames Room illusion [36]. Line segments can be backprojected to various combinations of orientations (one degree of freedom for each segment), and the combination that

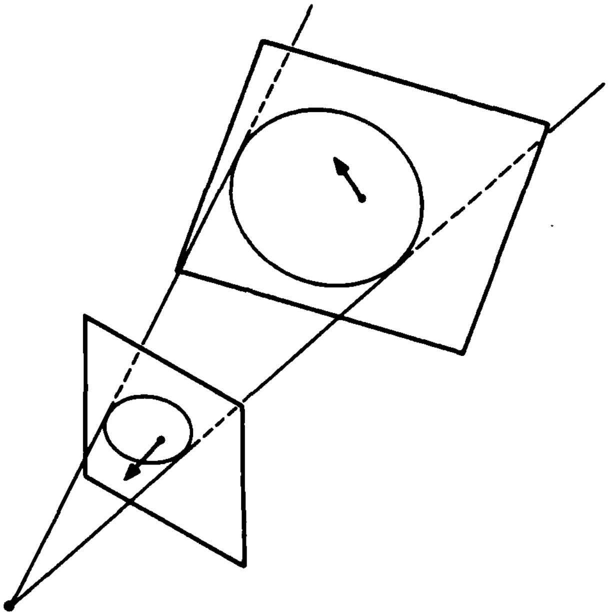


Figure 8: Planar Backprojection

leads to the most orthogonal basis for the vector space of the scene corresponds to the correct interpretation.

It is even possible to extend the concept of backprojection to include illumination and albedo models. The three forms of geometrical backprojection just described generate different shapes from one viewpoint. In addition to varying shape, one could, in principal, vary illumination (for example, by adding or moving point sources), or vary albedo, while satisfying the constraints imposed by the reflectance observed in the image. An image such as the Bumpy Torus (Fig. 2) could be explained in terms of a single-point-source illumination, a uniform albedo, and a smoothly curving surface; or it could be explained in terms of two point sources, implying a shape and/or an albedo that would be very complex. The choice is clear. The problem of using reflectance constraints effectively — connecting the surface shape and albedo to the observed reflectances — is difficult, but there has been promising recent work in this area [37].

In any realistic language, the number of possible encodings of any particular stimulus would likely be enormous. The task of enumerating all of them, while possible in principle, would be hopeless in practice. Information in the primitive encoding, however, may be used to suggest possible forms of final encodings. For example, "T-junctions" suggest occlusion, and sets of lines intersecting at a common point suggest parallelism. In this approach the role of local "cues" is merely to suggest descriptions, but the final interpretation depends only on the form of the descriptions and is not required to account for all the cues.

3.3. Levels of Description

Using the formalism of differential geometry, we can, in principle, represent 2D or 3D figures in a precise, well-founded, intuitive way that is independent of the choice of a coordinate system. Section 4 discusses in detail how the simplicity of figures can be estimated from descriptions. The method requires further descriptions at different levels of specificity. We will use the notation developed by Carnap [38].

We assume that a curve or surface has a precise description that captures all aspects of its shape. For example, in the case of smooth, continuous curves, these descriptions consist of analytic expressions for curvature and torsion. We denote a precise description of this form by D^{prec}.

We can convert precise descriptions to approximate **individual** descriptions (of which there are a finite number) by sampling over the parameter space at a certain precision of measurement. For example, a smooth, continuous curve can be sampled at intervals of arc to yield a sequence of curvature and torsion measurements (to some precision). Denote an individual description by D^{ind}, let N be the number of samples, and let K be the number of possible distinct measurements. That is, we divide the measurement space (of, for example, curvature and torsion) into K cells Q_j ($j = 1, \ldots, K$). [4]

Finally, we can convert an individual description to a **statistical** description by counting the number of elements N_j belonging to each cell. In other words, we can construct a histogram D^{st} from D^{ind}. The statistical description gives the frequencies of occurrences of the various measurements.

Each level of description is implied by its predecessor:

$$D^{\text{prec}} \Rightarrow D^{\text{ind}} \Rightarrow D^{\text{st}} .$$

Individual descriptions that imply the same statistical description are said to be *statistically equivalent*. A statistical description represents a disjunction of individual descriptions. The simplicity measure that will be described in Section 4 is based on the size of this set.

4. Why are Some Interpretations Preferred?

This approach to figural perception begins with 2D image descriptions that are disordered, or in which the implicit order is hidden, and, through backprojection, proceeds to construct consistent 3D descriptions that may be more ordered. In other words, it works from complex descriptions to simple ones. If 3D descriptions of very simple, highly ordered form are found, they are chosen as the best interpretations. The logical justification for selecting simple descriptions over complex ones is essentially the principle of Occam's Razor.

We can draw a loose analogy with a famous problem of physics. Statistical mechanics provides an explanation, based on probabilistic reasoning, of the behavior of irreversible thermodynamic processes, and, in particular, of the Second Law of Thermodynamics, which states that the entropy of a closed system must increase. In simple terms, closed systems invariably evolve from ordered states to less ordered ones. Boltzman [39] and Gibbs [40] invented the mathematical formalisms of statistical mechanics to account for this. The important insight was to identify entropy, which had hitherto been defined only in terms of macroscopic physical measurements, with probabilistic descriptions of the microscopic states of thermodynamic systems. They were able to show that, because the number of disordered states is vastly greater than the number of ordered ones, the probability of the system moving into a disordered state is extremely high. More recently, Prigogine [41] has further developed the thermodynamic concepts of structure and disorder of complex systems.

In a seminal paper that began the field of information theory, Shannon used the concept of entropy as a measure of information [42]. At first, this seemed to be a completely different concept than thermodynamic entropy, but Brillouin showed that they were closely connected

[4] A *finitization* of this sort happens when a discrete image is created.

and consistent [43], [44]. Jaynes showed that the thermodynamic concept could be derived from Shannon's measure [45], [46].

4.1. A Model of Structure and Information

The property that we use for selecting preferred descriptions is minimum entropy.

Entropy is defined for statistical descriptions, for individual descriptions by implication, and for precise descriptions under some system of finitization. Using the notation developed in Section 3.3, assume we have a statistical description D^{st} with cell numbers $N_1, \ldots, N_K$. The number of statistically equivalent individual descriptions D^{ind} with these cell numbers is given by

$$z(D^{st}) = \frac{N!}{N_1! \cdots N_K!} \; . \tag{3}$$

The minimum value of z occurs when all elements belong to the same cell (the homogeneous case):

$$z_{min} = 1 \; .$$

The maximum occurs when all cell numbers are as nearly equal as possible (the maximally heterogeneous case). Assuming that N is divisible by K:

$$z_{max} = \frac{N!}{(\frac{N}{K})!^K} \; .$$

A system with a statistical description of large z is more disordered than one with small z. This is because the statistical description of large z can be realized in relatively many ways, and it gives us relatively little information about the underlying precise description. On the other hand, if a statistical description has small z, there are few possible individual descriptions. This observation is the heart of the minimum-entropy principle for figural perception.

Various sources define entropy in different ways. Shannon, for example, uses the formula:

$$H = -\sum_{j=1}^{K} p_j \ln p_j \; , \tag{4}$$

which can be related to z, the number of statistically equivalent individual descriptions consistent with a D^{st}, as follows. We take the probabilities p_j to be the observed probabilities in a statistical description:

$$p_j = N_j / N \; .$$

Applying Stirling's formula to (3), we obtain

$$\ln z \approx -N \sum_{j=1}^{K} p_j \ln p_j \; , \text{ for large N.}$$

Therefore, from (4),

$$H \approx \frac{\ln z}{N} \; . \tag{5}$$

The important point is that entropy is always defined as a linear function of the logarithm of z, even though the details may differ from source to source. The base chosen for the logarithm will affect the units in which entropy is measured, of course, but, since we will only be concerned with comparisons of values, we can use any convenient base and treat entropy as a pure number.

The following definition, given by Carnap, has some useful properties:

$$S(D^{st}) = \ln z - N \ln K\ . \tag{6}$$

If N varies but the relative probabilities p_i do not change, then S is proportional to N. Furthermore, if each cell is divided into a fixed number q of new cells with equal cell numbers N_j/q, then S remains unchanged. These properties are computationally attractive because they allow entropies calculated for statistical descriptions with different N and K to be compared, which outweighs the minor inconvenience that $S \leq 0$.

The concept of entropy is notoriously opaque to intuition. The essential point is that a description will have high entropy when its elements occur with more-or-less the same probability, and it will have low entropy when a few measurements have much higher probabilities than all others. Shannon's measure, H, can be interpreted as the average amount of information per symbol in a description. An encoding is said to be *efficient* if its symbols occur with equal probability, and therefore carry equal amounts of information, or, equivalently, if the encoded description has maximum entropy. Shannon's original motivation was to discover how to use fixed-bandwidth communication channels most efficiently, and he was therefore led to the concept of entropy as a measure of the efficiency of coding schemes.

The **redundancy** of a description is defined as:

$$R = 1 - \frac{H}{H_{max}}\ , \tag{7}$$

or, in terms of Carnap's definition,

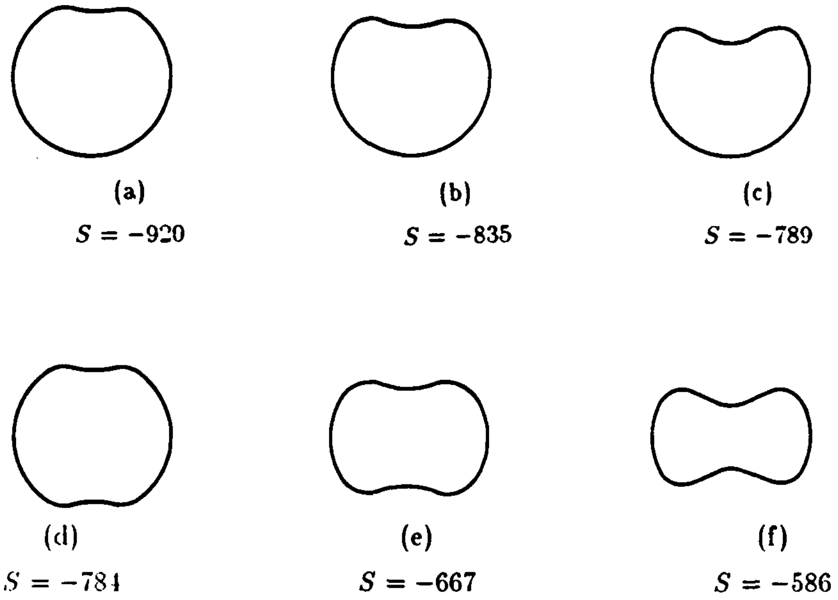
$$R = 1 - \frac{S + N \ln K}{S_{max} + N \ln K}\ . \tag{8}$$

Note that R is in the interval $[0, 1]$, and that $R = 0$ for an efficient encoding. An encoding with entropy significantly lower than the maximum possible value, however, will contain a degree of redundancy. Finding minimum-entropy interpretations is equivalent to finding maximally redundant ones. Redundancy is thereby *discovered* and can then be exploited to build more concise descriptions.

4.2. Some Examples

In this section a few simple examples of the inductive approach will be presented. The minimum entropy criterion will be applied to smooth, continuous, planar (zero-torsion) curves. We will show how various transformations affect the measured disorder of the curves.

Figures 9 to 12 show several curves created with cubic b-splines [47], which, in this case, comprise the precise descriptions of the figures. A cubic b-spline represents a smooth, continuous curve with a finite control polygon, which essentially determines the coefficients



(a) $S = -920$

(b) $S = -835$

(c) $S = -789$

(d) $S = -784$

(e) $S = -667$

(f) $S = -586$

Figure 9: Entropy under Change in Amplitude and Symmetry

of a cubic piecewise polynomial and which can, therefore, be used as an interpolation function [48]. An example of a control polygon is shown in Figure 11k.

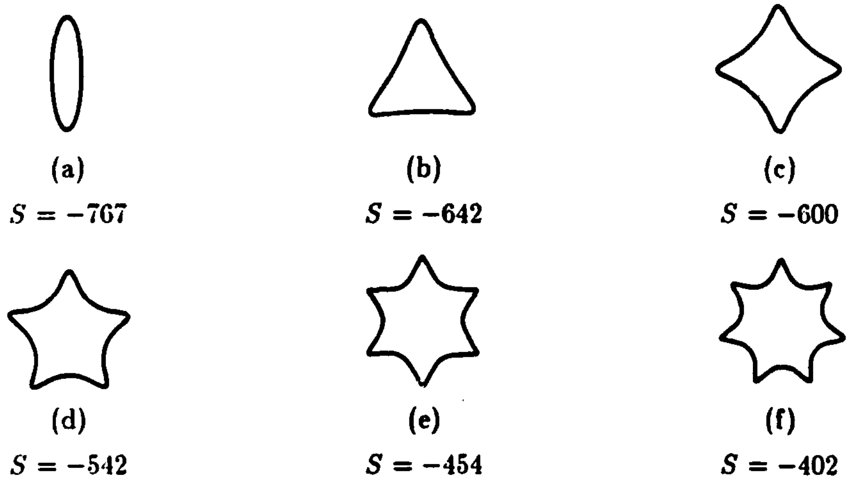
To make individual descriptions, the splines are sampled at a predetermined number N of equally spaced points (500 in all these examples), and curvature is determined analytically from the spline function. [5] A precision of measurement is then chosen (the parameter K, which was equal to 200 in all the examples).[6]

The first example (Figure 9) shows what happens to the entropy of an initially circular figure as its symmetry is broken, first into a series of three increasingly noncircular figures with one axis of symmetry ((a) through (c)), and then into a series of figures of the same amplitude as the first three, but with two axes of symmetry. Notice that, for a given symmetry, entropy increases monotonically with amplitude. Also, a two-fold symmetric figure has *higher* entropy, and, therefore, is less simple than a one-fold symmetric figure of comparable amplitude (e.g., compare (c) to (f)). This observation shows that Kanizsa's objection to coding theory mentioned in Section 2.2 does not apply to this method. More axes of symmetry do not imply more simplicity. Quite the contrary.

The next example (Figure 10) is another case of symmetry change. All the figures have the same amplitude and only differ by the number of lobes. Entropy monotonically increases with the number of lobes, or, in other words, figures with few axes of symmetry are judged to be simpler than comparable figures with many axes of symmetry. This behavior is quite surprising, because there is no explicit notion of symmetry built into the minimum-entropy

[5] If the precise spline function is not known *a priori*, curvature may be estimated by fitting circles to triplets of adjacent samples of the given figure. In either case, we can also relax the requirement that samples be equally spaced by keeping, as part of the description, the sequence of arc-length segments between unequally spaced samples. Entropy would then be computed using a two-part statistical description: one part for curvatures, and one for arc length.

[6] Before computing individual descriptions for a given set of curves, the interval of admissible measurements must also be fixed. If the bounds are set as tight as possible (i.e., to the actual minimum and maximum of all curvatures of the set of curves), the measurements will be as accurate as possible for a given K. The same bounds were used in all the examples.



(a) $S = -767$

(b) $S = -642$

(c) $S = -600$

(d) $S = -542$

(e) $S = -454$

(f) $S = -402$

Figure 10: Entropy under Change in Symmetry

model.

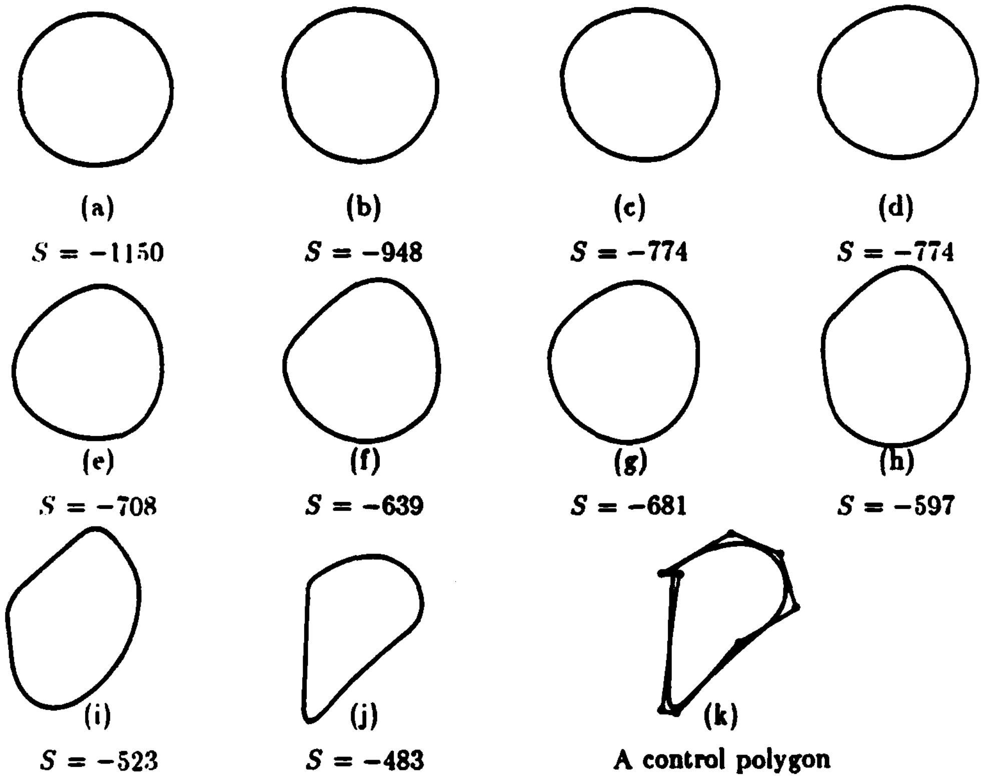
If we begin with a highly ordered curve and then introduce random changes, we would expect the curve to become more disordered: entropy should increase. Figure 11 shows that this is indeed the case. The eight vertices of the polygon used to generate an initially circular curve were perturbed by adding zero-mean gaussian noise. A sequence of curves was created by iterating this process. Each curve has undergone twice as many iterations as its predecessor. Entropy increases with the number of iterations — not monotonically, because of the random nature of the experiment (iteration (g) had lower entropy than iteration (f)), but as a statistical trend.

The final example (Figure 12) shows how the minimum-entropy principle can be used to select 3D *interpretations*. The curve in Figure 9c was rotated in azimuth and elevation and then projected in perspective. The resulting curve, shown in Figure 12a, was backprojected onto several hypothetical planes, which are indicated by tilted circles in the other figures. Just as in the previous examples, individual and statistical descriptions were computed for each of the backprojected figures, and their entropies were determined. As expected, the best interpretation has the lowest entropy, because it corresponds to the interpreted curve that is most regular.

4.3. Discussion

The minimum-entropy principle for figural perception expresses a preference for figures that are *simplest* in a certain sense. The measure of simplicity — negative entropy — can be interpreted in several ways, using metaphors of physics, information theory, and inductive reasoning.

Simplicity is the obverse of disorder, which is measured by entropy. Closed physical systems dissolve into disorder; which is to say, they undergo irreversible thermodynamic change. Perceptual systems are not closed, of course. They can freely exchange energy with their supporting systems, and thereby evolve into more ordered states. In a sense,



(a) $S = -1150$

(b) $S = -948$

(c) $S = -774$

(d) $S = -774$

(e) $S = -708$

(f) $S = -639$

(g) $S = -681$

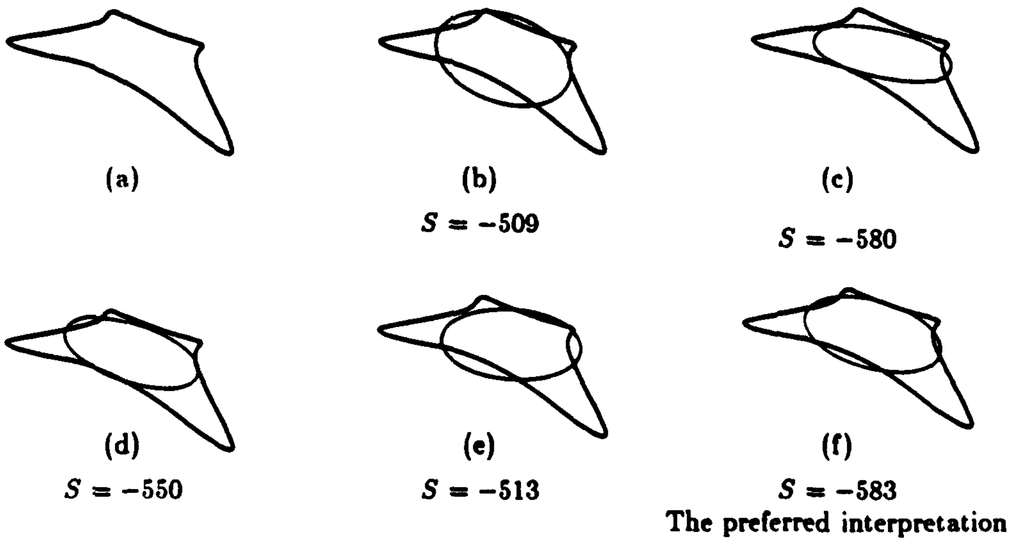
(h) $S = -597$

(i) $S = -523$

(j) $S = -483$

(k) A control polygon

Figure 11: Entropy under Random Perturbation



(a)

(b) $S = -509$

(c) $S = -580$

(d) $S = -550$

(e) $S = -513$

(f) $S = -583$ The preferred interpretation

Figure 12: Entropy under Backprojection

the minimum-entropy concept treats perception as the conceptual reversal of physically irreversible processes. Prigogine has developed the concept of entropy exchange to analyze the behavior of open systems [41].

In communication theory, the entropy of a message source is determined by the probabilities of the messages it sends. If there are many more-or-less equally probable messages (high entropy), the receiver is initially in a condition of high uncertainty; if there are relatively few, highly probable messages (low entropy), the receiver has less uncertainty. After receiving the message, the receiver gains an amount of information equal to the uncertainty that is resolved. There are two ways of measuring the amount of information in a message: (a) reduce the message to the shortest possible encoding (i.e., a nonredundant encoding) and then count the number of symbols, or (b) estimate the entropy directly from observed frequencies using Equation (6) and apply Formula (8). The coding theory discussed in Section 2 uses the first method, while the minimum entropy approach uses the second. The advantage to the second method is that it eliminates the need to actually construct a nonredundant encoding — a task that may require considerable cleverness. If we have two individual descriptions with distinct statistical descriptions (but with the same N, K, and bounds), and if one description has lower entropy than the other, then it is more redundant and can, in principle, be encoded with fewer symbols.

The entropic model of complexity, uncertainty, and disorder has profoundly influenced the mathematical foundations of inductive reasoning [49], [38], [50]. The first principle in this foundation has been called the principle of insufficient reason; namely, if there is insufficient reason to believe that several possibilities have different probabilities, one should behave as though they were equally probable. Using entropy as a measure of disorder or as a measure of information follows this principle for the following reason. Given a statistical description, all statistically equivalent individual descriptions are treated as equally probable:

$$P(D_i^{\text{ind}}) = \frac{1}{z(D^{\text{st}})} \; .$$

If we must choose from a variety of plausible interpretations with different statistical descriptions (e.g., as determined by backprojection), we choose the one leading to the most probable individual descriptions; that is, the one with the lowest entropy.

5. Conclusions

The inductive approach suggests a new direction for computational vision. We must face the fact that perception is not veridical and that deductive methods are therefore not appropriate for general-purpose vision. At the same time, approaches that rely on matching specific prior models are unsatisfactory, because they cannot explain the perception of abstract figures of which we have no prior experience, knowledge, or expectation. Recent work toward theories involving a so-called 2.5D sketch (see [51]), when considered as an explanation of figural perception, suffers from the same defect as the deductive approach: there is, in general, insufficient information in a single image to construct iconic, viewer-centered representations of physical surface properties. Relatively direct modes of perception, such as stereo and optic flow, may yield to this approach, but the interpretation of single images will not. Even stereo and optic flow require heuristic assumptions, such as the rigidity constraint, that are closely related to the information-theoretic concept of simplicity.

Induction seems to be a natural paradigm for human intelligence. By observing events, one recognizes correlations, and infers symmetry, causality, family resemblances, and other relationships. To be sure, the inferences may be wrong, but that's too bad. People make mistakes. In fact, one of the weaknesses of deduction is that it does not permit one to draw conclusions that may be in error (assuming the axioms are correct), but that represent the best conclusions under the circumstances.

Only a very small part of a full inductive theory of intelligence is presented in this paper, and several important questions remain to be addressed. For example, one can imagine hierarchies of descriptions, embedded in successively more concise, more global, and more idiosyncratic encoding schemes. To give a trivial example, a curve in the shape of the United States could be encoded as a sequence of arc lengths and curvatures, but it could also be encoded — much more concisely — as a reference to a known shape. How might these hierarchies of descriptions be structured, and how can efficient encoding schemes be learned through experience?

Bibliography

[1] Newman, W. M. and Sproull, R. F., **Principles of Interactive Computer Graphics**, McGraw-Hill, New York, 2nd Ed., 1979.

[2] Mackworth, A. K., How to see a simple world: an exegesis of some computer programs for scene analysis. In *Machine Intelligence 8*, pp. 510-537, N. L. Collins and D. Michie, Eds., American Elsevier, New York, 1977.

[3] Waltz, D., Understanding line drawings of scenes with shadows, in **The Psychology of Computer Vision**, P. H. Winston, ed., McGraw-Hill, New York, 1975.

[4] Kanade, T., A theory of origami world, *Artificial Intelligence* **13** (1980), 279-311.

[5] Rock, I., **The Logic of Perception**, M.I.T. Press, Cambridge, Massachusetts (1983).

[6] Huffman, D. A. Impossible objects as nonsense sentences, in: B. Meltzler and D. Mitchie, (Eds.), *Machine Intelligence* **6** pp. 295-323, 1971.

[7] Mackworth, A. K., Interpreting pictures of polyhedral scenes, *Artificial Intelligence* **4** (1973) 121-137.

[8] Kanade, T., Recovery of the 3-D shape of an object from a single view, *Artificial Intelligence*, **17** (1981) 409-460.

[9] Kender, J., Shape from texture, Ph.D. Thesis, Carnegie-Mellon University, Computer Science Department, November 1980.

[10] Ikeuchi, K., Shape from regular patterns (an example of constraint propagation in vision), A.I. Memo 567, M.I.T., Artificial Intelligence Laboratory, March 1980.

[11] Barnard, S. T., Interpreting perspective images, *Artificial Intelligence* **21** (1983), 435-462.

[12] Witkin, A. P., Recovering surface shape and orientation from texture, *Artificial Intelligence* **17** (1981) 17-45.

[13] Brady, M. and A. Yuille, An extremum principle for shape from contour, *Proceedings of IJCAI-83*, 969-972, Karlsruhe, West Germany, August 8-12, 1983.

[14] Grimson, W.E.L., A computational theory of visual surface interpolation, *Phil. Trans. R. Soc. Lond., B*, **298**, 217-253.

[15] Grimson, W.E.L., An implementation of a computational theory of visual surface interpolation, *Computer Vision, Graphics, and Image Processing*, **22**, 39-69.

[16] Grimson, W.E.L, Surface consistency constraints in vision, *Computer Vision, Graphics, and Image Processing*, **24**, 28-51.

[17] Terzopoulos, D., Multi-Resolution Computation of Visible-Surface Representations, PhD Dissertation, Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology, January 1984.

[18] Attneave, F., **Applications of Information Theory to Psychology**, Henry Holt and Co., New York (1959).

[19] Fitts, P. M., Stimulate correlates of visual pattern recognition, *Journal of Experimental Psychology*, 1957, **70**, 21-37.

[20] Garner, W. R., **Uncertainty and Structure as Psychophysical Concepts**, New York: Wiley, 1962.

[21] Garner, W. R., **The Processing of Information and Structure**, Lawrence Erlbaum Associates, Potomac, Maryland (1974).

[22] Hake, H. W., Contributions of Psychology in the Study of Pattern Vision, W.A.D.C. Technical Report 57-62, Dayton, Ohio: Wright Air Development Center, 1957.

[23] Klemmer, E. T., Perception of linear dot patterns, *Journal of Experimentai Psychology*, 1963, **65**, 468-473.

[24] Quastler, R., **Information Theory in Psychology**, Glencoe, Ill.: Free Press, 1955.

[25] Buffart, H., Leeuwenberg, E., and Restle, F., Coding theory of visual pattern completion, *Journal of Experimental Psychology: Human Perception and Performance*, Vol. 7, No. 2, April 1981, 241-274.

[26] Kanizsa, G., **Organization in Vision**, Praeger, New York (1979).

[27] Borisenko, A. I. and Tarapov, I. E., **Vector and Tensor Analysis**, Dover, New York, 1968.

[28] Lipschutz, M. M., **Differential Geometry**, McGraw-Hill, New York, 1969.

[29] Barnard, S. T. and Pentland, A. P., Three-dimensional shape from line drawings, *Proceedings of IJCAI-83*, 1062-1064, Karlsruhe, West Germany, August 8-12, 1983.

[30] Stevens, K. A., The visual interpretation of surface contours, *Artificial Intelligence* **17** August, 1981, 47-73.

[31] Kirkpatrick, S., C. D. Gelatt, Jr., and M. P. Vecchi, Optimization by simulated annealing, *Science*, Vol. 220, No. 4598, May 13, 1983, 671-680.

[32] Barrow, H. G. and Tenenbaum, J. M., Interpreting line drawings as three-dimensional surfaces, *Artificial Intelligence* **17** August, 1981, 75-116.

[33] Barnard, S. T., Choosing a basis for perceptual space, Technical Note 315, Artificial Intelligence Center, SRI International, Menlo Park, California, January 1984.

[34] Attneave, F. and Frost, R., The determination of perceived tridimensional orientation by minimum criteria, *Perception & Psychophysics*, vol. 6, no. 6B, December, 1969, 391-396.

[35] Perkins, D., Visual discrimination between rectangular and nonrectangular parallelopipeds, *Perception & Psychophysics*, vol. 12, no. 5, 1972, 396-400.

[36] Ittelson, W. H., **The Ames Demonstrations in Perception**, Hafner Publishing Company, New York, 1968.

[37] Koenderink, J.J. and van Doorn, A.J., Photometric invariants related to solid shape, *Optica Acta*, 1980, vol. 27, no. 7, 981-996.

[38] Carnap, R., **Two Essays on Entropy**, University of California Press, Berkeley, 1977.

[39] Boltzman, L., Weitere Studien über das Wärmegleichgewicht unter Gasmolekülen. Sitzungsber. d. königl. Akad. d. Wiss., Vienna: 1872. *Reprinted in* Boltzman: Wissenschaftliche Abhandlungen. Leipzig: Barth, 1909, vol. 1.

[40] Gibbs, J. W., *Elementary Principles in Statistical Mechanics*, Reprinted in Collected Works and Commentary, Yale University Press (1936), and by Dover Publications, Inc. (1960).

[41] Prigogine, I., Unity of physical laws and levels of description, in **Interpretations of Life and Mind**, M. Greene (Ed.), Humanities Press, New York, 1971.

[42] Shannon, C., *Bell System Tech. J.*, **27**, 379-623. Reprinted in C. E. Shannon and W. Weaver, **The Mathematical Theory of Communication**, University of Illinois Press, Urbana (1949).

[43] Brillouin, L., Negentropy principle of information, *Journal of Applied Physics*, **24**, N9, 1152-63, September 1953.

[44] Brillouin, L., **Science and Information Theory**, Academic Press, New York, 1962.

[45] Jaynes, E. T., Information theory and statistical mechanics, *Physical Review*, **106**, 620, 1957.

[46] Jaynes, E. T., Information theory and statistical mechanics, *Physical Review*, **108**, 171, 1957.

[47] Gordon, W. J. and Riesenfeld, R. F., B-spline curves and surfaces, in **Computer Aided Geometric Design**, Robert E. Barnhill and Richard F. Riesenfeld (eds.), Academic Press, 1974, 95-126.

[48] Barsky, B. A , The Beta-Spline: A Local Representation Based on Shape Parameters and Fundamental Geometric Measures, PhD dissertation, Department of Computer Science, University of Utah, December, 1981.

[49] Christensen, R., **Foundations of Inductive Reasoning**, published by R. Christensen, Berkeley, 1964.

[50] Levine, R.D. and Tribus, M. (Eds.), **The Maximum Entropy Formalism**, M.I.T. Press, Cambridge (1979).

[51] Marr, D., **Vision**, W. H. Freeman, San Francisco (1982).

Appendix G

Object Labeling Using Generic Knowledge

By: Pascal V. Fua and Andrew J. Hanson

Object Labeling Using Generic Knowledge

By: Pascal V. Fua
Andrew J. Hanson

Artificial Intelligence Center
SRI International

SUMMARY: Substantial progress has been made on an investigation into methods of intelligent feature extraction. The initial task is the extraction of man-made structures from aerial imagery. We merge pixel-level techniques with geometric reasoning and generic (as opposed to specific or template-like) object descriptions. Methods are proposed for identifying, explaining, and compensating for expected verifiable discrepancies between the generic models and the image data.

1 Introduction

Our purpose here is to describe the current state of our investigations into the problem of extracting features from aerial imagery using computer-automated methods. In order to limit the scope of the work to achievable implementation goals, we have adopted the following conditions and assumptions:

- **Object type:** We restrict ourselves to the identification of man-made structures in aerial imagery, thereby providing the opportunity to use such observations as the presence of straight lines to focus attention on regions likely to be components of a target object.

- **Initial data:** We assume that as initial data we are given a digitized aerial image that is essentially a straight-down view. In addition, we assume we are provided with a syntactic partition of that image; typically, we use a partition computed by an Ohlander-style segmenter such as that provided by Laws [SRI Technical Note 334].

- **Knowledge characteristics:** We avoid template-based and feature-space techniques by using *object-generic* knowledge in our analysis. In particular, we concentrate on generic models of target objects that are based on the way people would describe the process of recognizing an instance of the target class of objects in an aerial image.

A reasonable goal of future extensions would be to remove some of these restrictions and investigate broader classes of problems.

The focal points of the current work are the following:

- Smoothly integrating pixel-level information with geometric reasoning techniques and generic object descriptions.

- Developing the ability to characterize, explain, and correct discrepancies between generic models and the data extracted from the image, with particular emphasis on understanding the nature of anomalies in the initial scene partition.

In the following sections, we present the general features of our current approach to the feature extraction problem, along with some sample results and remarks about future objectives.

2 Approach to the Feature Extraction Problem

Our basic approach to feature extraction using object-generic knowledge involves the sequence of steps outlined below:

Select generic shapes. Correlation techniques based on explicit shape templates, Hough transform methods, and other statistical methods of extracting shape from imagery suffer from a wide variety of well-known failings. Objects such as "a house," or "a chair" can take on a rich variety of forms that no template-based method can adequately cope with. Humans utilize generic knowledge about forms and constraints to identify members of object classes; it is our intention to mimic the human recognition process by using generic relations and descriptive forms similar to those used by people.

A simple example of what we mean by generic knowledge about "a house" would be the following:

- Houses have central regions bounded by 4 or more straight lines usually meeting at right angles.
- Houses cast shadows and the shadows often have straight boundaries.
- Houses have yards and driveways that may have straight borders confusable with the house itself.
- Some houses have peaked roofs, porches, or gables that separate the actual house into two or more house-like regions.
- Houses are arranged in simple geometric patterns near streets. Many houses have driveways and sidewalks connecting them to the street.

Select Elementary Object Features. Each individual object type will have certain characteristic features that can be exploited. In the case of houses and many other cultural objects, there exist very straight lines with distinctive relationships. Using a geometric reasoning system employing spatial relationships between lines and regions, we can build up evidence for certain classes of objects. For example, areas enclosed by three perpendicular lines forming a "U" are highly likely to belong to a roof of a house, although other hypotheses must also be investigated and evaluated.

Search for Object Instances. Given the initial image, a syntactically-based segmentation, and an edge operator, we can extract generic features, discover those implying the possible presence of target objects, and produce regions that are candidates for the desired object type in the image.

Invoke Knowledge Base of Processes. Once tentative identifications are made, they must be verified. The next step of our procedure is to access a knowledge base of the processes that a human expert might know were likely to produce a false identification. In some sense, this knowledge base is equivalent to a set of *explanations* coupled with appropriate low-level procedures that can actually test and verify or reject a given explanation.

The explanations can serve several purposes:

- Cause decisions to be made based upon available models and upon high-level information about the given scene and the overall task goal.
- Explain to a human observer the nature of the decision process that is taking place so that the human can feel confident that appropriate results are being generated.
- Provide a human implementor with insights into the weaknesses and strengths of the rule and explanation procedure, thus allowing improvements in the process to be made quickly and accurately.

Below we give an example of the sorts of rules currently being used to understand and explain the appearance of a set of three parallel lines that might be a peaked roof; this is typical of the classes of phenomena that need further understanding before information can be successfully utilized in higher-level identification procedures.

```
(if   (or (peaked-roof-house L1 L2 L3)
          (or (shadow-and-roof L1 L2 L3)      ; (L1 L2) is the shadow
              (shadow-and-roof L3 L2 L1))     ; (L3 L2) is the shadow
          (or (house-and-yard L1 L2 L3)       ; (L1 L2) is the house
              (house-and-yard L3 L2 L1))      ; (L3 L2) is the house
          (non-house-cultural-source L1 L2 L3))
 then (explainable-parallel-lines L1 L2 L3))

(if   (and (roof-like-enclosed-region L1 L2)
           (roof-like-enclosed-region L2 L3)
           (or (peaked-reflectance-agreement L1 L2 L3)
               (peaked-reflectance-agreement L3 L2 L1))
 then (peaked-roof-house L1 L2 L3)

(if   (and (shadow-like-enclosed-region L1 L2)
           (roof-like-enclosed-region L2 L3))

 then (shadow-and-roof L1 L2 L3))

(if   (and (roof-like-enclosed-region L1 L2)
           (yard-like-enclosed-region L2 L3))

 then (house-and-yard L1 L2 L3))

(if   (and
        (or (yard-like-enclosed-region L1 L2)
            (sidewalk-like-enclosed-region L1 L2)
```

```
                (driveway-like-enclosed-region L1 L2)
                (street-like-enclosed-region L1 L2))
            (or (yard-like-enclosed-region L2 L3)
                (sidewalk-like-enclosed-region L2 L3)
                (driveway-like-enclosed-region L2 L3)
                (street-like-enclosed-region L2 L3)))
    then (non-house-cultural-source L1 L2 L3))
```

Here the symbols L1, L2, L3 represent symbolic instances of the line elements being tested, and (explainable-parallel-lines L1 L2 L3) is the fundamental goal that would be tested by the knowledge engine until it came to rest on some root-level piece of knowledge like (shadow-like-enclosed-region L1 L2) that was either prestored or dynamically computed by a specially-coded procedure.

Refine Scene Partition. We recall that the image analysis process began with a digitized image and a segmentation. The segmentation process itself must be understood and characterized well enough so that a rule base of corrections like the one described above can be generated and meaningfully utilized. By combining knowledge of the generic object models with knowledge of the likely behavioral anomalies of the initial segmentation, we can generate an improved segmentation containing rearranged image segments that bear appropriate labels. These labeled regions are the extracted features that satisfy the goal of our efforts.

3 Process Outline and Sample Analysis

In this section we take a sample aerial image containing houses and carry out an analysis using the software currently implemented.

A schematic outline of the software organization is provided in Figure 1. The fundamental knowledge-based components of the system on the left of the diagram are:

- Spatial Relationships. This subsystem contains knowledge about geometric relationships among lines and regions. The relationships include parallelness, perpendicularity, enclosure, and linkability (including calculation of the line linking the two initial lines).

- Object Descriptions. This subsystem contains relational descriptions of the target objects, e.g., how lines combine to describe a figure that is house-like in a generic fashion.

- Anomaly Explanations. This subsystem contains descriptions of the types of errors that are liable to be made by the initial segmentation and object identification system, along with procedures to generate testable hypotheses and check them to discover the true (or at least more likely) objects. The system also provides explanations and motivations for its own actions.

The central boxes in Figure 1 represent the flow of data during the analysis procedure, beginning at the bottom with the image data itself, and proceeding to object labeling as the final symbolic result.

We now present some figures giving an example of a typical analysis process. In Figure 2, we show the initial scene; in Figure 3, we overlay the initial segmentation boundaries on image.

Next, in Figure 4, we show the dominant (locally) straight edges found in the scene. These edges provide strong evidence of cultural objects near the indicated locations.

Finally, in Figures 5 and 6, we show two candidate houses that were identified solely by the use of geometric reasoning on the adjacency of regions to shapes (such as "U" and "L") that are very likely to be found enclosing portions of houses in the segmentation. No corrective or explanatory reasoning was done to achieve this result. Later, we expect to be able to handle more complex images when the reasoning and explanation capabilities have been added.

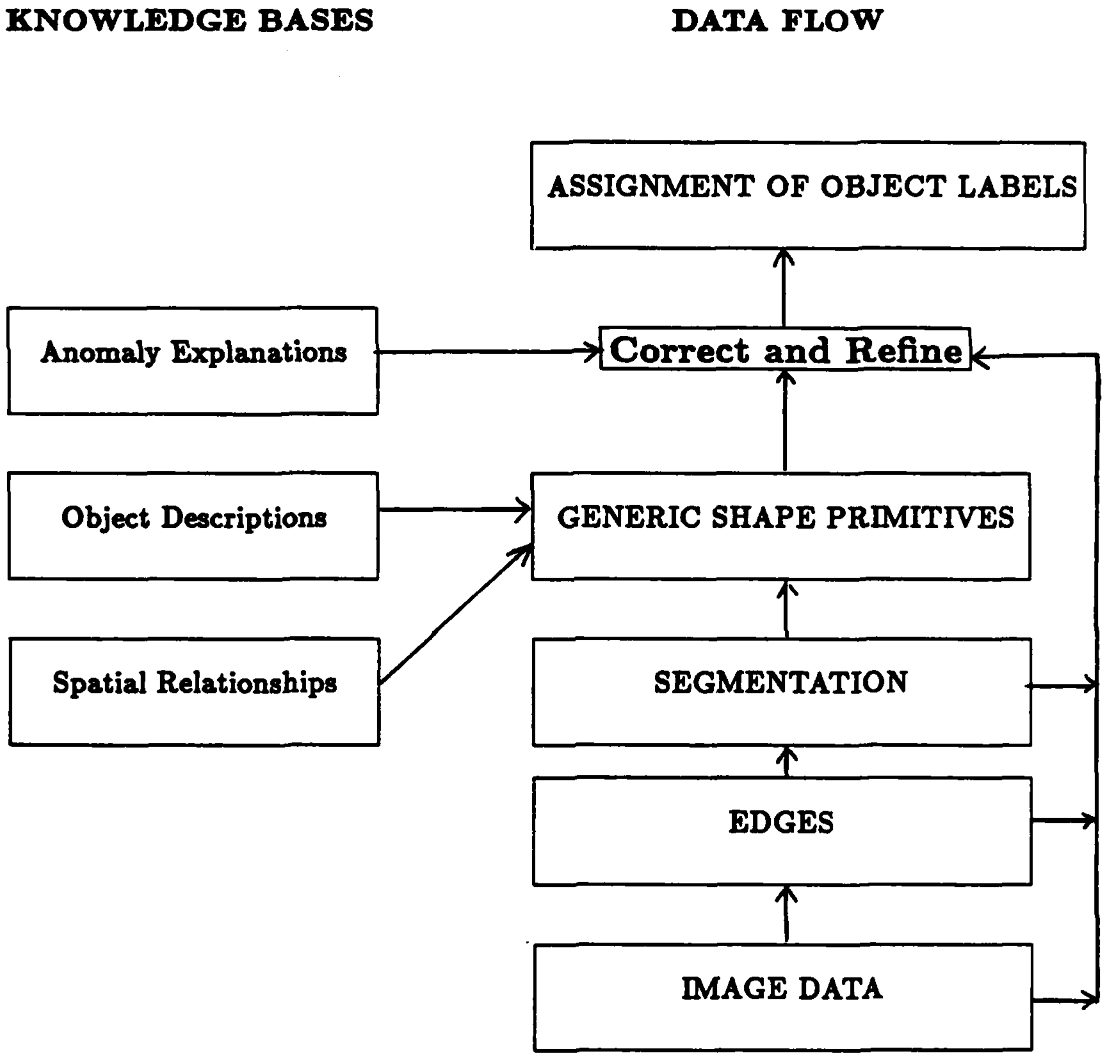


Figure 1: Program structure and data flow for the feature extraction process.

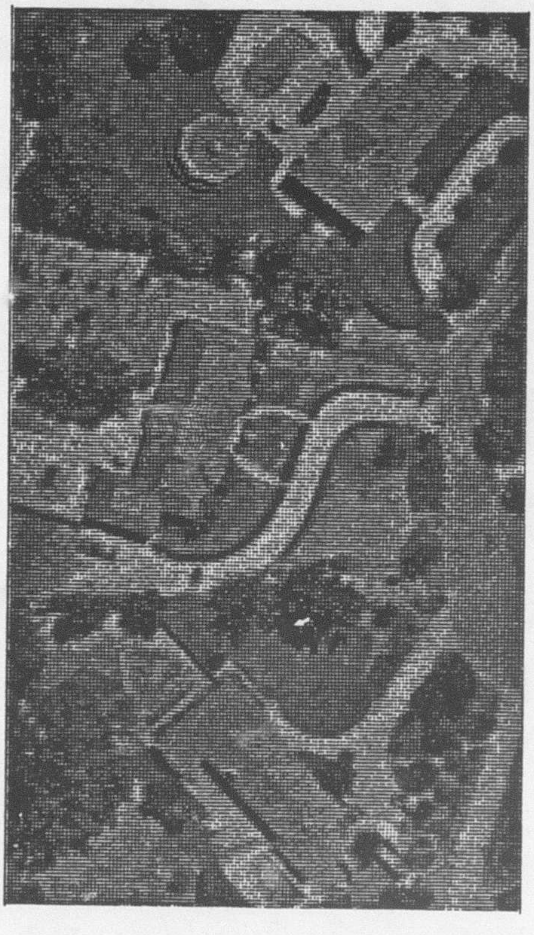


Figure 2: Half-toned representation of an aerial scene including houses.

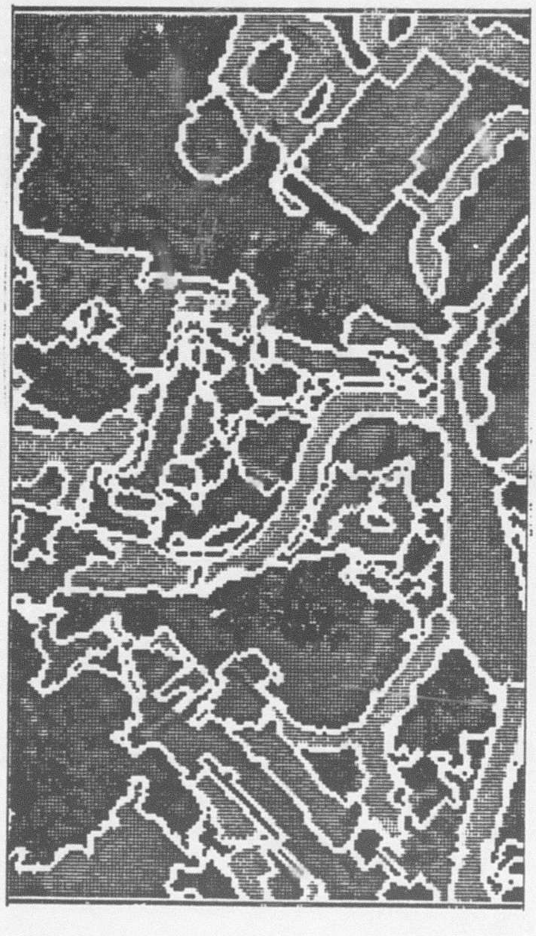


Figure 3: Scene with Ohlander-style segmentation overlaid.

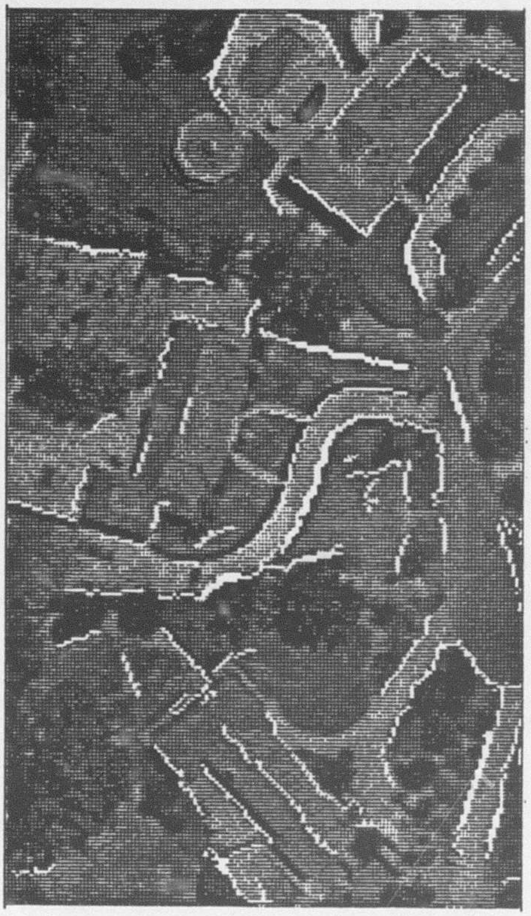


Figure 4: Scene with the best straight edges (indicating possible cultural object nearby) overlaid.

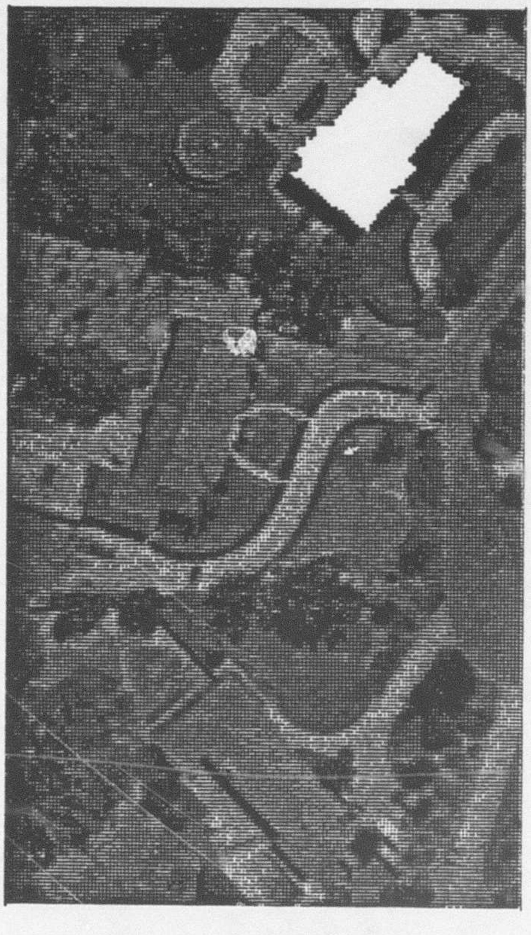


Figure 5: An identified house-like region. The result is based on elementary house-component shapes.

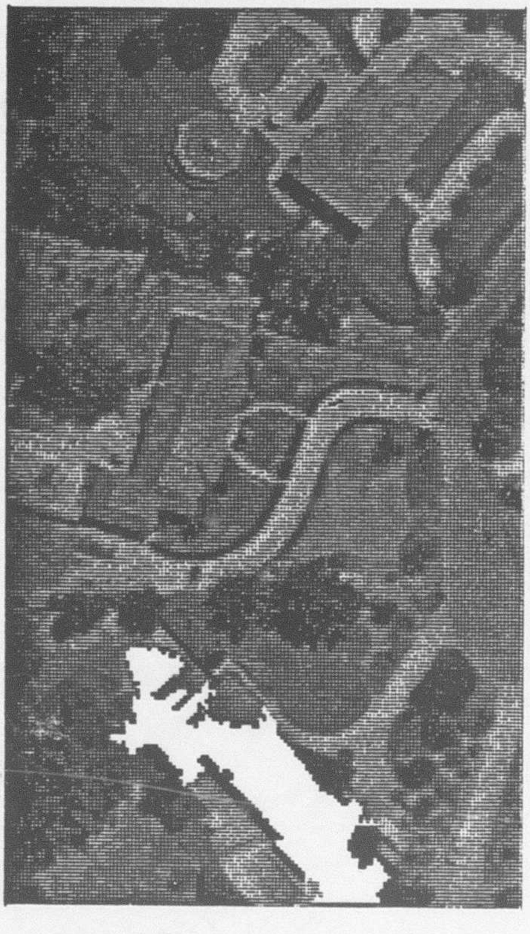


Figure 6: A set of regions that combine into a larger region that includes a house-like shape.

4 Interactive user interface

This system will eventually be one of a series of "Interactive Expert Cartographic Systems," that will support an interactive problem-solving scenario involving guidance from the human operator. A typical scenario will involve steps such as the following:

- Task definition. User will provide background and goal information, context description, and training examples. When possible, the user will carry out interactive modeling of target object characteristics and of the problem solution process.
- Model-based recognition. System will carry out the task of recognizing objects based on stored models, returning to the user on occasion for conflict resolution and confirmation that the strategy chosen is yielding the desired results.
- Explanation. The system will maintain a current database of explanations; explanations can be provided continuously or supplied upon request of the user. The logic of the processes carried out will be available for examination and immediate correction if deficiencies are detected.
- Evaluation and correction. The final stage of the process includes evaluation of the success of the results, together with the option of repeating the analysis with different input guidance, refining partial results, or performing major additions to the rule base and remedial procedure library.

5 Future directions

Some of the initial enhancements that we plan to add are the following:

- Generalize capabilities to allow many classes of target objects.
- Replace the current procedural encoding of low-level algorithmic steps and geometrical reasoning by a more flexible rule-based approach.
- Include the ability to use additional high-level information such as sun angle and camera angle to predict shadows and perspective distortion of target shapes. Use these features as additional factors in the object identification process.
- Support exploitation of multiple images covering the same scene.

Appendix H

Perceptual Organization and the Representation of Natural Form

By: Alex P. Pentland

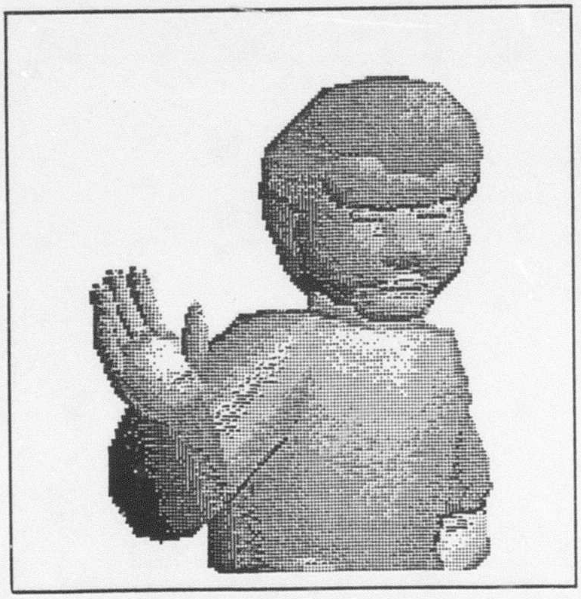
PERCEPTUAL ORGANIZATION AND THE REPRESENTATION OF NATURAL FORM

SRI Technical Note 357

July 15, 1985

By: Alex P. Pentland, Computer Scientist

Artificial Intelligence Center
Computer Science and Technology Division



PERCEPTUAL ORGANIZATION AND

THE REPRESENTATION OF NATURAL FORM

Alex P. Pentland

Artificial Intelligence Center, SRI International
333 Ravenswood Ave, Menlo Park, CA 94025
and
Center for the Study of Language and Information
Stanford University, Stanford CA 94038

1 Introduction

Our world is very highly structured: evolution repeats its solutions whenever possible [1], and inanimate forms are constrained by physical laws to a limited number of basic patterns [2]. The apparent complexity of our environment is produced from this limited vocabulary by compounding these basic forms in myriad different combinations. Indeed, the highly patterned nature of our environment is a necessary precondition for intelligence; for if the apparent complexity of our environment were approximately the same as its intrinsic [Kolmogorov] complexity then intelligent prediction and planning would be impossible, for there would be no lawful relations. It is this internal structuring of our environment, then, that causes object features to cluster into groups, and allows us to reason sucessfully using the simplified category descriptions that we typically employ [3].

To support our reasoning abilities, therefore, perception must recover these environmental regularities — e.g., rigidity, "objectness", axes of symmetry — for later use in cognitive processes. This recovery of structure is known as *perceptual organization*, familiar from such research efforts as the Gestalt movement [4], Johansson's [5] study of the organization of motion perception, and more recently Marr and Nishihara's [6,7] theory of form perception using a description based on generalized cylinders [8]. The problem of perceptual organization is important because the structural regularities that perception recovers are the parts from which we construct our picture of the world; they are the building blocks of all cognitive activities.

To understand how our perceptual apparatus can produce meaningful cognitive building blocks from the unstructured array of image intensities, we want a representation that both correctly models important environmental regularities [9,10] and also accounts for the perceptual organization we impose on the stimulus — the one structuring of the stimulus that we know can support general-purpose cognitive activity. Unfortunately, the representations that are currently available were originally developed for other purposes (e.g.,

This research was made possible by a grant from the National Science Foundation, Grant No. DCR-83-12766, in part by Defense Advanced Research Projects Agency contract no. MDA 903-83-C-0027, and in part by a grant from the Systems Development Foundation.

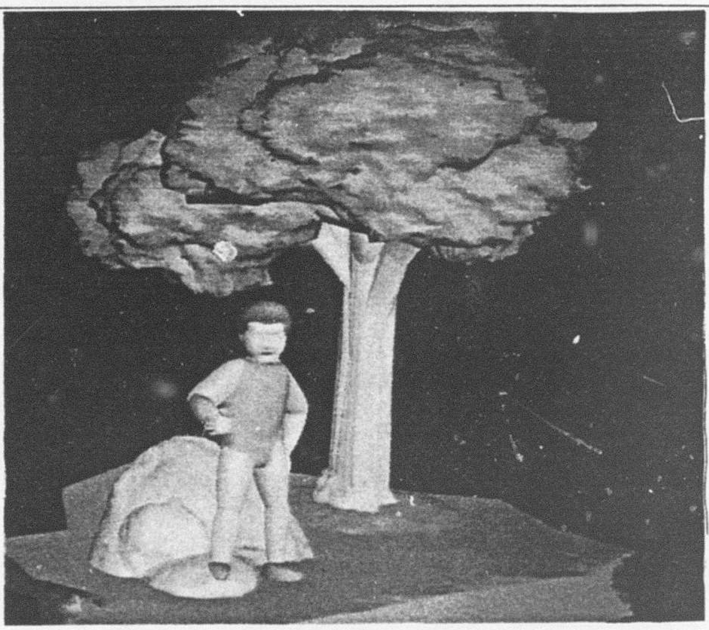


Figure 1. A scene described and generated by the representational system described within: tree leaves and bark, rocks and hair are fractal surfaces, the overall shape is described by Boolean combination of appropriately deformed superquadrics. Only 56 primitives are required (fewer than 500 bytes of information) to specify this scene. The slightly cartoon-like appearance is primarily due to the lack of surface texturing.

the point-wise descriptions of physics or the platonic-solids descriptions of engineering) and are therefore often unsuitable for the problems of perception.

Most current-day vision research, for instance, is based on the point-wise representation used in describing the physics of image formation, and consequently research has focussed on analyzing image content on a local, point-by-point basis. Biological visual systems, however, can not recover scene structure from such local information[1] In fact, biological visual systems are are strikingly insensitive to the point-by-point particulars of the image formation process (e.g., reflectance function or illuminant direction), factors that figure prominently in todays best vision research.

Rather than depending only upon point-wise information, people seem to make heavy use of the larger-scale structure of the scene in order to guide their perceptual interpretation. Similarly, the performance of most current-day vision algorithms depends critically upon assumed larger-scale structural context, e.g., upon assuming smoothness or isotropy. To progress towards general-purpose vision, therefore, we need new representations capable of describing these critical larger-scale structures; the "parts" or "building blocks" that we use to organize the image and provide a framework for perceptual interpretation.

Towards this end Marr and Nishihara [6] proposed a scheme using hierarchies of cylinderlike modeling primitives to describe natural forms. Their proposal is, it seems,

[1] As you can confirm for yourself by looking through a long, one-inch wide tube such as found in rolls of wrapping paper.

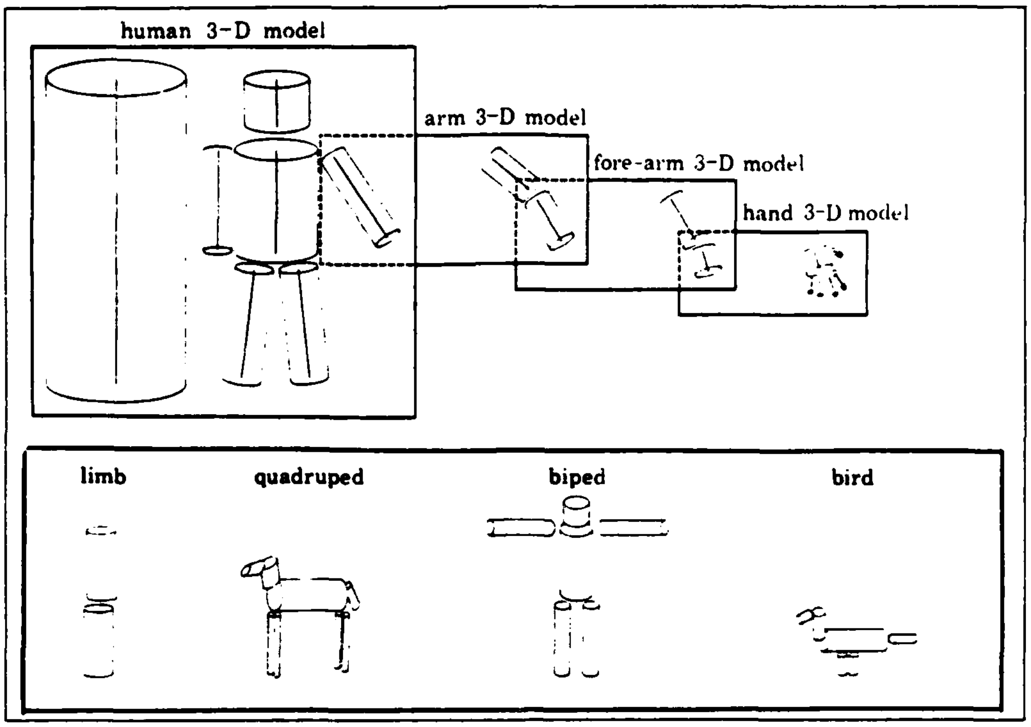


Figure 2. Marr and Nishihara's scheme for the description of biological forms.

the most widely known representation suggested to date; it captures many of our intuitions about axes of symmetry and hierarchical description [see also Agin and Binford (11), Nevatia and Binford (12), Badler and Bajacsy (13) and Brady (14)]. Further, in recent years representations like theirs have found considerable success in industrial-style machine vision systems where an exact model of the specific objects that are to be discovered in the image data is available [15,16]. Unfortunately, such a representation is only capable of an extremely abstracted description of most natural and biological forms, as is illustrated in Figure 2. It cannot accurately and succinctly[2] describe most natural animate forms or produce a succinct description of complex inanimate forms such as clouds or mountains.

In this paper we will present a representational system — indeed, a logic — that has proven competent to accurately describe an extensive variety of natural forms (e.g., people, mountains, clouds, trees), as well as man-made forms, in a succinct and psychologically natural manner. Figure 1 shows an example of a scene described in this representation; only 56 descriptive "parts" (about 500 bytes of information) were employed. We will then present evidence that we can use the special properties of this representational system to recover descriptions of specific objects from image data, and finally we will argue that these recovered descriptions are extremely useful in supporting both commonsense reasoning and man-machine communcation.

[2] If we retreats from cylinders to generalized cylinders we can, of course, describe such shapes accurately. The cost of such retreat is that we must introduce a 1-D (at least) function describing the sweeping function; which makes the representation neither succinct nor intuitively attractive.

2 Vision, Cognition, and Models of Scene Structure

Perception is the mind's window on the world: its task is to recognize and report objects and relations that are important to the organism. It is this perceptual link between the *objective* environment and our *conception* of the environment that makes our thoughts meaningful; that ensures that they have some correspondence with the surrounding world.

Because the objects and relations recovered by perception are the primitive predicates upon which all cognition is built, the particular way in which our perceptual apparatus organizes sensory data — that is, which regularities are noted and which are ignored — places strong constraints on the ways in which we can think about our environment. When perception organizes the sensory data in a way unsuited to the current task even simple problems can become nearly impossible to solve, as is illustrated by problems where you "see" the solution only when you "look" at them in the right way.

Understanding how to identify the important regularities and relate them to the primitive elements of cognition is, consequently, the principal goal of research into visual function. The central problem in such research, of course, is that the sensory data underdetermines the scene structure. Image pixels, by themselves, can determine nothing. Some knowledge of image formation and of how the world is structured is *required* in order to obtain *any* assertion about the viewed scene.

Visual perception, therefore, is best viewed as the process of recognizing image regularities that are known — on the basis of one's model of the world — to be reliably related to cognitive primitives. The need for a model cannot be sidestepped, for it is the model that relates the theory's representations and computations to the state of the real world, and thus explains the semantics — the *meaning* — of the theory. A theory of visual function that has no model of the world also has no meaning[3] .

Understanding the early stages of perception as the interpretation of sensory data by use of models (knowledge) of the world has, of course, been a standard vision research paradigm. To date, however, most models have been of two kinds: high-level, *specific* models, e.g., of people or houses, and low-level models of image formation, e.g., of edges. The reason research has almost exclusively focused on these two types of model is a result more of historical accident than conscious decision. The well-developed fields of optics, material science and physics (especially photometry) have provided well worked out and easily adaptable models of image formation, while engineering, especially recent work in computer aided design, have provided standard ways of modeling industrial parts, airplanes and so forth.

Both the use of image formation models and specialized models has been thoroughly investigated. It appears to us that both types of model, although useful for many applications, encounter insuperable difficulties when applied to the problems faced by, for instance,

[3] Theories of visual function, therefore, are based on models: models of how the world is structured and of how this structure is evidenced by regularities in the image. Much vision research is *not* model based, of course: research on the mechanisms of vision (e.g., parallel processors, neurons), or on procedures for accomplishing visual tasks (e.g., variational calculus, relaxation methods) need not employ models of the world. But to understand visual *function* — that is, how one can infer information about the world — it is necessary to have a model of the salient world structure and of how that structure evidences itself in the image. Only then can one understand how certain features of the image can allow recognition and recovery of the information of interest.

a general purpose robot. In the next two subheadings we will examine both types of model, outline the advantages and disadvantages in using these models for recovering important scene information, and then in the remainder of this section motivate, develop and investigate an alternative category of models.

2.1 Models of Image Formation

Most recent research in computational vision has focused on using point-wise models of image formation borrowed from optics, material science and physics. This research has been pursued within the general framework originally suggested by Marr [10] and by Barrow and Tenenbaum [17], in which vision proceeds through a succession of levels of representation. The initial level is computed directly from local image features, and higher levels are then computed from the information contained in small regions of the preceding levels. Processing is primarily data-driven (i.e., bottom-up).

In Marr's scheme the initial level is called the "raw primal sketch," and contains a description of significant local image structure, e.g., edges, lines, or flow field vectors, represented in the form of an array of feature descriptors that preserves the local two-dimensional geometry of the image. The second level is called the "2 1/2D sketch," and is intended to describe local surface properties (e.g., color, orientation) and discontinuities in a viewer-centered coordinate frame. Again, the recovered local surface properties are placed in a set of numeric arrays in registration with original image. From this point an object-centered, volumetric representation was to be computed, such as is illustrated by Figure 2. The rationale for this level of representation is that tasks such as navigation or object recognition seem to require discription in a viewpoint-independent coordinate frame.

Despite its prevalence, there are serious problems that seem to be inherent to this research paradigm. Because scene structure is underdetermined by the local image data [18], researchers have been forced to make *unverifiable* assumptions about large-scale structure (e.g., smoothness, isotropy) in order to derive useful information from their local analyses of the image. In the real world, unfortunately, such assumptions are often seriously in error: in natural scenes the image formation parameters change in fairly arbitrary ways from point to point, making any assumption about local context quite tenuous. As a result those techniques that rely on weak, general assumptions such as isotropy have proved fragile and error-prone, while those that rely on strong assumptions such as smoothness are simply not applicable to many natural scenes.

That such difficulties have been encountered should not, perhaps, be too surprising. It is easily demonstrated (by looking through a viewing or reduction tube) that people can obtain little information about the world from a local image patch taken out of its context. It is also clear that detailed, analytic models of the image formation process are not essential to human perception; humans function quite well with range finder images (where brightness is proportional to distance rather than a function of surface orientation), electron microscope images (which are approximately the reverse of normal images), and distorted and noisy images of all kinds — not to mention paintings and drawings.

Perhaps even more fundamentally, however, even if depth maps and other maps of intrinsic surface properties could be reliably and densely computed, how useful would they be? As can be seen from industrial vision work [16] using laser range data, a depth map is still basically an image. Although useful for obstacle avoidance and other very simple tasks, it still must be segmented, interpreted and so forth before it can be used for any more sophisticated task. The conclusion to be drawn from such work is that image-like measurements of range and other surface properties contribute incrementally, in much the same way as color: they add a dimension that simplifies some decisions, but they do *not* solve the difficult problems encountered in image interpretation (for a more extended discussion of image formation models see Witkin and Tenenbaum [19]).

2.2 Specialized Models

The alternative to models of image formation has been engineering-style representations; e.g., CAD-CAM models of specific objects that are to be identified and located. Such detailed, specific models evidence themselves in image data in an extremely complex manner, in part because the models themselves are often complex, but more importantly because it is the objects' surface shape, and not the appearance of the object, that is described. As the object's orientation varies, therefore, these models produce a *very* large number of different pixel configurations — to say nothing of what happens when we vary the illumination and imaging conditions. As a consequence, the image regularities that allow reliable recognition across all of the allowable configurations are very subtle and complex.

The large number of possible appearances for such models makes the problem of recognizing them very difficult — unless an extremely simplified representation is employed. The most common type of simplified representation is that of a wireframe model whose components correspond to the imaged edges. Such a simplified representation permits reliable recognition of models with currently available computational resources, given that we are in a restricted environment where the descriptive power of such wireframe models is sufficient, e.g., as in industrial applications. As a result systems based on CAD-like models of specific objects have provided most of the success stories in machine vision.

Despite this success, the use of an impoverished representation generally means that the flexibility, reliability and discriminablity of the recognition process is limited. Thus research efforts employing specific object models have floundered whenever the number of objects to be recognized becomes large, when the objects may be largely obscured, or when there are many unknown objects also present in the scene.

An even more substantive limitation of systems that employ *only* high-level, specific models is that there is no way to learn new *types* of object: new model types must be specially entered, usually by hand, into the database of known models. This is a significant limitation, because the ability to encounter a new type of object, enter it into a catalog of known objects, and thereafter recognize it is an absolute requirement of truly general purpose vision.

2.3 Part and Process Models

Some sort of additional constraint is required to overcome the fundamental problem of insufficient information being available from the image. If sufficient constraint is not available from models of image formation, then from where? Human vision seems to function quite well as long as the imaging process preserves the basic spatial structure of the scene; we are able to perceive electron microscope images, depth images, line drawings and so forth. It seems, therefore, that human perception must be exploiting constraints provided by the structure of the scene without reliance on quantitative, point-wise models of the image formation process. What is required, then, are models of scene structure that capture something about the larger-scale structure of our environment. We cannot, however, appeal to CAD-like models of specific objects because of the impossibility of learning new descriptions.

In response to these seemingly intractable problems some researchers have begun to search for a third type of model, one with a grain size intermediate between the point-wise models of image formation and the complex, specific models of particular objects [20]. There is good reason to believe that it may be possible to accurately describe our world by means of such intermediate-grain models; that world can be modeled as a relatively small set of generic processes that occur again and again, with the apparent complexity of our environment being produced from this limited vocabulary by compounding these basic forms in myriad different combinations.

We have known for over a century that evolution repeats its solutions whenever possible [1], resulting in great regularities across all species: there are but a few types of limb, a few types of skin, a few types of leaf, and a few patterns of branching. An amazingly good model of a tree, for instance, is the composition of a simple branching process with three-dimensional texture processes for generating bark and leaves [21]; the same branching models can also serve for rivers, veins, or coral. Similarly, it is now being discovered that inanimate forms may also be constrained by physical laws to a limited number of basic patterns [2,22]. Mandelbrot has shown that such apparently complex forms such as clouds, hills, coastlines or cheese can all be described by simple patterns recursively repeated at all different scales [22], while Stevens presents strong evidence that natural textures occur in but a few basic forms [2].

Indeed, such internal structure in our environment is a necessary precondition for intelligence; for if the apparent complexity of our environment were approximately the same as its intrinsic [Kolmogorov] complexity then intelligent prediction and planning would be impossible, for there could be no lawful relationships [23]. It is exactly this internal structuring of our environment that causes object features to cluster into groups, and allows us to sucessfully employ "commonsense reasoning," i.e., to reason by use of the simplified category descriptions that we typically employ [3].

It appears, then, that it may be possible to accurately describe the world in terms of *parts*: macroscopic models that, in relatively simple combination, can be used to form rough-and-ready models of the objects in our world and how they behave. If we adopt this view, then the central problems of perception are *not* how to describe images, surfaces, and volumes, so that we may eventually arrive at recognition of high-level models [10]. Rather, the central problems for perception are to find a set of generically applicable part-models, describe how they combine to form images, and then use this description in order to recognize the content of an image as a combination of these generic primitives. This new proposal, then, is to dispense entirely with initial stages of description and begin immediately with recognition of parts models: models that are in principle much like models of houses and chairs, but that are more generally applicable and less detailed.

Because such models would be simpler than models of specific objects we would expect that we could be more readily characterize how they would appear in an image. On the other hand, because they describe larger-scale structure than point-wise models of image formation, we would expect that they might not suffer from the problems of underdetermination that have forced researchers to make unrealisticly strong assumptions such as smoothness or isotropy. Besides offering a good balance between complexity and reliablity, such intermediate-grain parts models spark considerable interest because they describe the world in the right terms: they speak qualitatively of whole objects and of relations betweeen objects, rather than of local surface patches or of specific objects. Thus they can potentially provide a vocabulary for describing the world at the grain size that is most often directly useful to us.

The problem with forming such "parts" models is that they must be complex enough to be reliably recognizable, and yet simple enough to reasonably serve as building blocks for specific object models. Current 3-D machine vision systems, for instance, typically use rectangular solids and cylinders to model specific shapes. Using these primitives for the automatic construction of a description for an arbitrary new object has not proven possible, except[4] (as in industrial or urban imagery) when the set of objects that will be encountered is constrained to be simple combinations of rectangular solids or cylinders [24]. To support truly general purpose vision, therefore, we need to develop new modeling primitives that can be used to build descriptions of arbitrary objects, and that are recognizable in standard imagery. Our work towards this goal is the subject of the remainder of this paper.

3 A Representation For Natural Forms

We present here a representational system — indeed, a logic — that has been proven competent to accurately describe an extensive variety of natural forms (e.g., people, mountains, clouds, trees), as well as man-made forms, in a succinct and natural manner (see Figure 1). The idea behind this representational system is to provide a vocabulary of models and operations that will allow us to model our world as the relatively simple composition of component "parts," parts that are reliably recognizable from image data.

The most primitive notion in this represention may be thought of as a "lump of clay," a modeling primitive that may be deformed and shaped, but which is intended to correspond roughly to our naive perceptual notion of "a part." It is worth noting that this notion of "part" corresponds roughly with that used by Konderink and Van Doorn [25,26] or by Hoffman and Richards [27] in their analysis of line drawings. For this basic modeling element we use a parameterized family of shapes known as a *superquadrics* [28,29], which are described (adopting the notation $\cos\eta = C_\eta$, $\sin\omega = S_\omega$) by the following equation:

[4] A caveat should be noted with respect to laser rangefinders and the like: in some cases the thousands of range measurements provided by these active sensors can give enough additional constraint to allow recovery of low-level, polygon-like descriptions of novel objects.

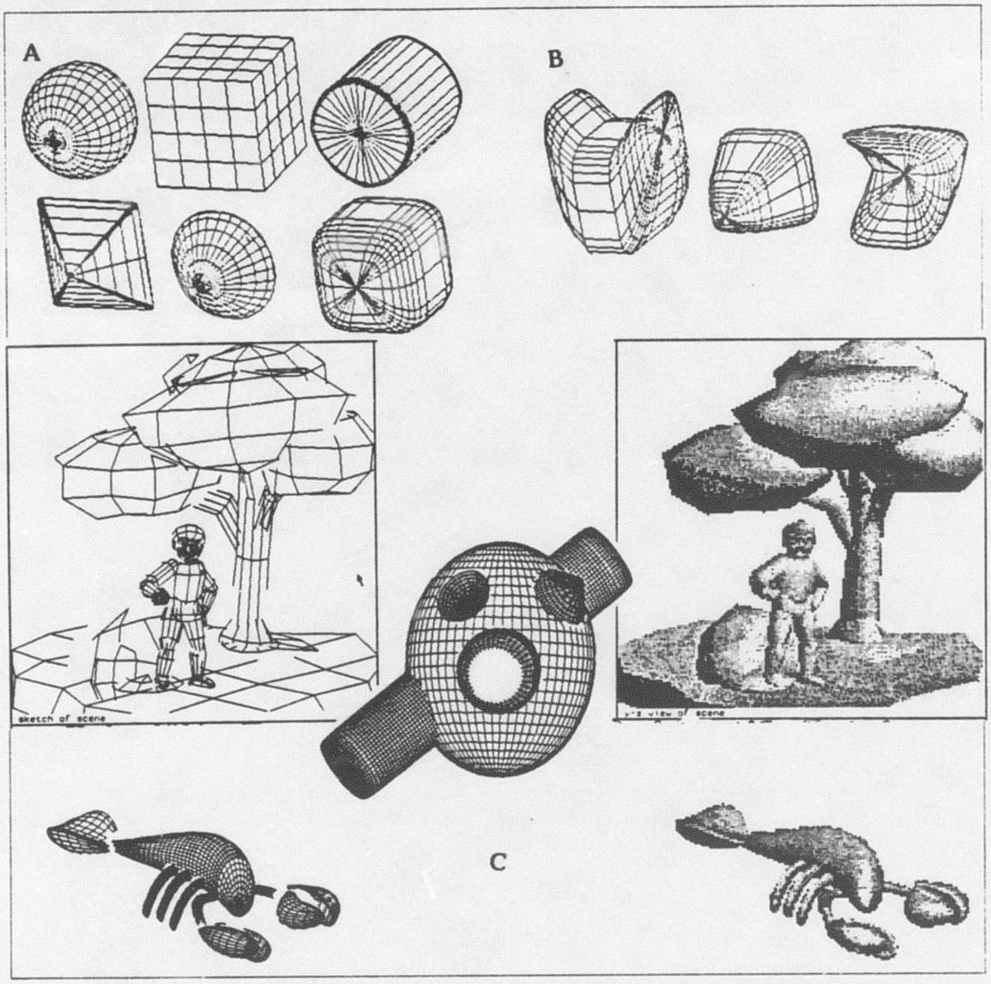


Figure 3. (a) A sampling of the basic forms allowed, (b) deformations of these forms, (c) Boolean combination (or's and nots) of the basic forms.

$$\mathbf{X}(\eta,\omega)=\begin{pmatrix} C_\eta^{\epsilon_1}C_\omega^{\epsilon_2} \\ C_\eta^{\epsilon_1}S_\omega^{\epsilon_2} \\ S_\eta^{\epsilon_1} \end{pmatrix}$$

where $\chi(\eta,\omega)$ is a three-dimensional vector that sweeps out a surface parameterized in latitude η and longitude ω, with the surface's shape controlled by the parameters e_1 and e_2. This family of functions includes cubes, cylinders, spheres, diamonds and pyramidal shapes as well as the round-edged shapes intermediate between these standard shapes. Some of these shapes are illustrated in Figure 3(a). Superquadrics are, therefore, a superset of the modeling primitives currently in common use.

These basic "lumps of clay" (with various symmetries and profiles) are used as prototypes that are then deformed by stretching, bending, twisting or tapering, and then combined using Boolean operations to form new, complex prototypes that may, recursively, again be subjected to deformation and Boolean combination. As an example, the back of a

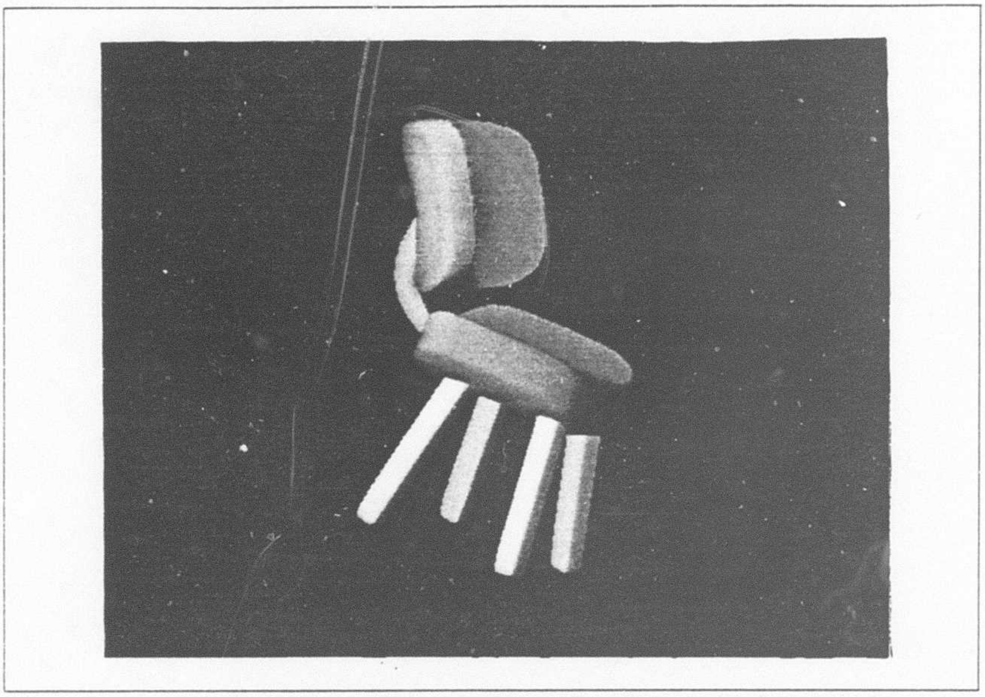


Figure 4. A chair formed from Boolean combinations of appropriately deformed superquadrics.

chair is a rounded-edge cube that has been flattened along one axis, and then bent somewhat to accommodate the rounded human form. The bottom of the chair is a similar object, but rotated 90°, and by "anding" these two parts together with elongated retangular primitives describing the chair legs we obtain a complete description of the chair, as illustrated in Figure 4.

This descriptive language is designed to describe shapes in a manner that corresponds to a possible formative history, e.g., how one would create a given shape by combining lumps of clay. We have found that by using such a process-oriented, possible-history representation we force the resulting descriptions to group points that have similar causal histories, thus obtaining "parts" that interact with the world in a relatively simple, holistic manner. This further simplifies many reasoning tasks, because the parameters and components that affect interactions tend to be explictly represented rather than being some complex or difficult-to-calculate function of the descriptions' variables. For instance, use of this type of representation sufficiently simplifies questions about spatial relationships, intersection, image appearance, and so forth that we have been able to use it to construct a real-time 3-D graphical modeling system, using a Symbolics 3600 computer[5] . This system, called "SuperSketch," was used to make the figures in this paper.

[5] "Real-time" in this case means that an "lump" can be moved, hidden surface removal accomplished, and drawn as a 100 polygon line drawing approximation in 1/8th of a second, and a complex, full color image such as Figure 1 can be rendered in approximately 20 seconds. The Symbolics speed is roughly comparable to a VAX 11/780, except for being almost an order of magnitude slower on the floating point operations that are used heavily in this modeling system.

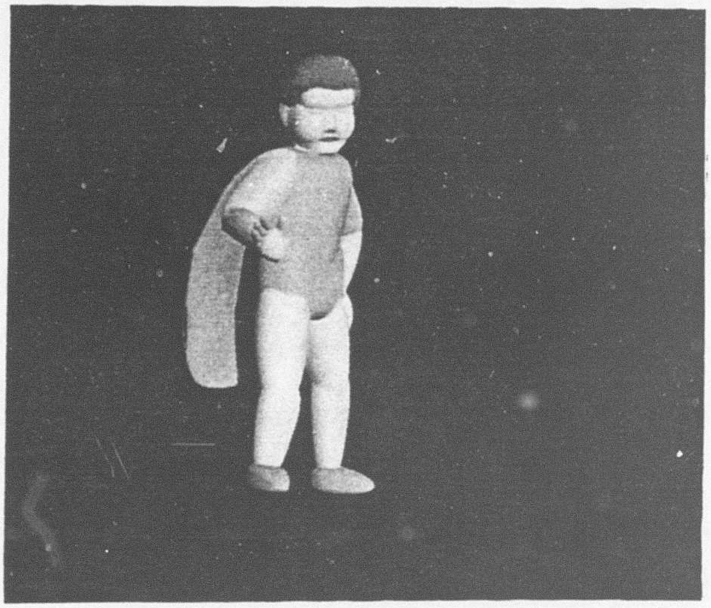


Figure 5. The human form described (and rendered) by use of this representational system; only 40 primitives are required, approximately 300 bytes of information.

Such descriptions may be written as a predicate calculus formula. We may then use this description, which has a clear model-theoretic semantics, in conjunction with constraint satisfaction or theorem-proving mechanisms, to accomplish whatever reasoning is required. Interestingly, it has been found that when adult human subjects are required to describe imagery verbally with completely novel content, their typical spontaneous strategy is to employ a descriptive system analogous to this one — i.e., form is described by modifying and combining prototypes [30]. The classic work by Rosch [3] supports the view that such a prototype-and-differences descriptive system is common in human reasoning: she showed that even primitive New Guinea tribesmen (who appear to have no concept of regular geometric shapes) form the geometric prototypes in much the same manner as people from other cultures and describe novel shapes in terms of differences from these prototypes.

This representational system provides a grammar of form that has surprising descriptive power. Such descriptions have the intuitively satisfying nature of the Marr and Nishihara scheme; they incorporate hierarchies of primitives with axes of symmetry. This new descriptive language, however, is considerably more powerful than other representations that have been suggested. For example, a trivial comparison is that we can describe a wider range of basic shapes, as shown in Figure 3(a). By allowing deformations of these shapes we greatly expand the range of primitives allowed, as shown in Figure 3(b) (see also Barr [31], Hollerbach [32], Leyton [42] on describing shape using modifications of prototypes). We have, so far, required only stretching, bending, tapering and twisting deformations to construct an extremely wide variety of objects. But the most powerful notion in this language is that of allowing [hierarchical] Boolean combination of these primitives. This intuitively

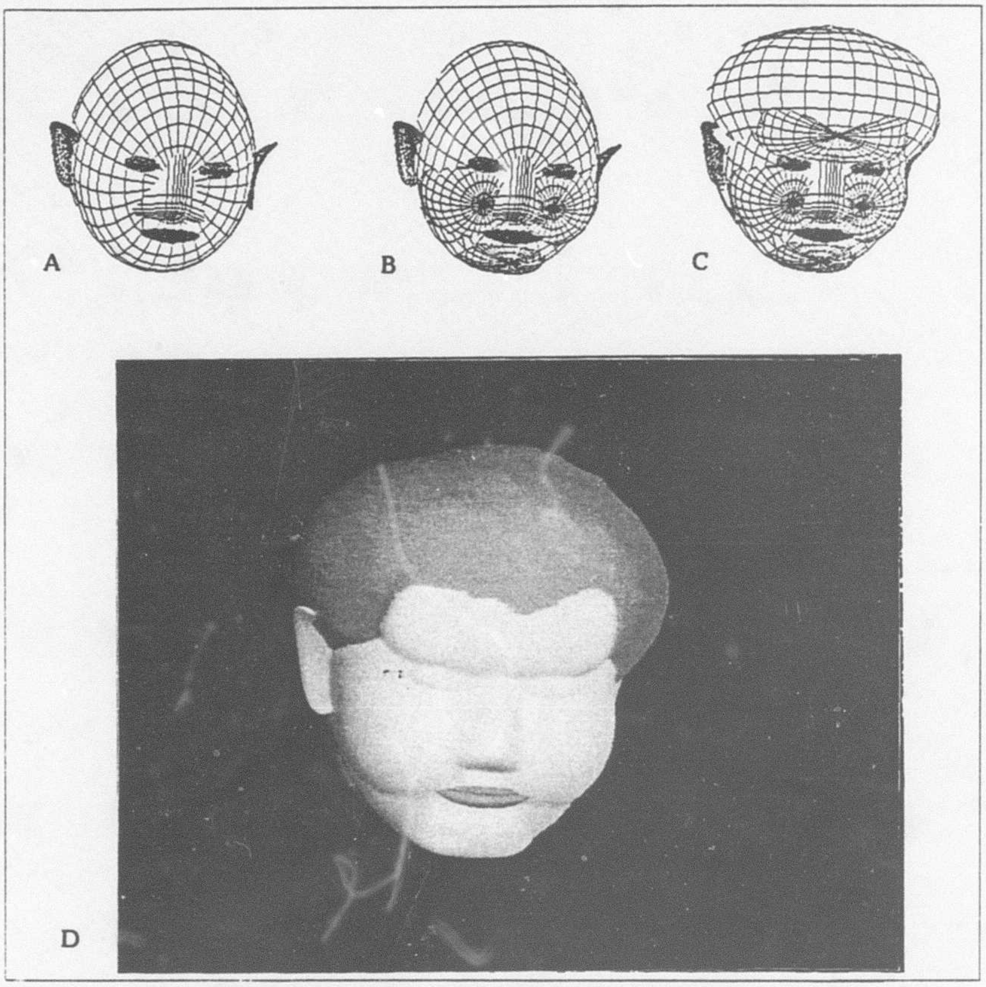


Figure 6. Figure 6(a) shows that the basic form for the head is a slightly tapered ellipsoid; to this basic form is added a somewhat cubical nose, bent pancake-like primitives for ears, bent thin ellipsoids for lips, and almond-shaped eyes. Figure 6(b) show the addition of rounded cheeks and slightly pointed chin (is this Yoda from Star Wars?), and finally Figure 6(c) shows the addition of a squarish forehead and slightly fractalized hair. The smoothly shaded result is shown in Figure 6(d) — it is a reasonably accurate human head, composed of only 13 primitives, specified by slightly less than 100 bytes of information.

attractive constructive solid modeling approach — building specific object descriptions by applying the logical set operations "and", "or" and "not" to component *parts* — introduces a language-like generative power that allows the creation of a tremendous variety of form, such as is illustrated by Figure 3(c) or by Figure 1.

3.1 Biological forms

Biological forms such as the human body are naturally described by hierarchical Boolean combinations of the basic primitives, allowing the construction of accurate — but quite simple — descriptions of the detailed shape, as illustrated by Figure 5 (the slightly cartoon-like nature of these illustrations is due primarily to the lack of surface texturing). The entire human body shown in Figure 5, including face and hands, requires combining only 40 primitives, or approximately 300 bytes of information (these informational requirements are not a function of body position). Similarly, the description for the face requires the combination of only 13 primitives, or fewer than 100 bytes of information. The extreme brevity of these descriptions makes many otherwise difficult reasoning tasks relatively simple, i.e., even NP-complete problems can be easily solved when the size of the problem is small enough.

In Figure 5 (as in all cases examined to date) when we try to model a particular 3-D form we find that we are able to describe — indeed, we are almost *forced* to describe — the shape in a manner that corresponds to the organization our perceptual apparatus imposes upon the image. That is, the components of the description match one-to-one with our naive perceptual notion of the "parts" in the figure, e.g., the face in Figure 5 is composed of primitives that correspond exactly to the cheeks, chin, nose, forehead, ears, and so forth. Figure 6 shows how the face is formed from the Boolean sum of several different primitives. The basic form for the head is a slightly tapered ellipsoid. To this basic form is added a somewhat cubical nose, bent pancake-like primitives for ears, bent thin ellipsoids for lips, and almond-shaped eyes, as is shown in Figure 6(a). Figure 6(b) show the addition of rounded cheeks and a slightly pointed chin (is this Yoda from Star Wars?), and finally Figure 6(c) shows the addition of a squarish forehead and slightly fractalized hair. The smoothly shaded result is shown in Figure 6(d) — it is a reasonably accurate human head, composed of only 13 primitives, specified by slightly less than 100 bytes of information. One should remember that this representation is *not* in any way tailored for describing the human form: it is a general-purpose vocabulary.

The correspondence between the organization of descriptions made in this representation and human perceptual organization is important because it is strong evidence that we are on the right track. The fact that the distinctions made in this representation are very similar to those made by people makes it likely that descriptions couched in this language will be useful in a wide variety of commonsense reasoning tasks, e.g., that the vocabulary of this representation might constitute a good set of primitive predicates for the Naive Physics [33] research program[6] . Similarly, the ability to make the right "part" distinctions offers hope that we can form qualitative descriptions of specific objects ("Ted's face") or of classes of objects ("a long, thin face") by specifying contraints on part parameters and on relations bewteen parts, in the manner of Winston [46,47] or of Davis [48].

And, of course, such representational correspondence is also important because it provides the basis for useful man-machine interaction.

[6] Descriptions that correspond to a possible formative history explicitly group together parts of a form that have a similar causal history, i.e., that came about in the same manner. It appears that such groupings have a strong tendency to *continue* to act as a simple whole. Why this should be true is unclear; perhaps there are only a few basic categories of physical interaction that all may be characterized using the same definition of "part."

3.2 Complex inanimate forms

Many naturally occuring forms are fractals[7] [22,34-36]; Mandelbrot, for instance, shows that fractal surfaces are produced by several basic physical processes. One general characterization of naturally occuring fractals is that they are the end result of any physical processes that randomly modifies shape through local action, i.e., they are a generalization of random walks and Brownian motion. After innumerable repetitions, such processes will typically produce a fractal surface shape. Thus clouds, mountains, turbulent water, lightning and even music have all been shown to have a fractal form.

During the last two years we have developed these fractals into a model for describing complex, natural surface shapes [34,35,37], and have found that it furnishes a good description for many such surfaces. Evidence for the descriptive adequacy of this model comes from several sources. Recently conducted surveys of natural imagery [34-36], for instance, have found that this model accurately describes how most homogeneous textured or shaded image regions change over scale (change in resolution). The prevalence of surfaces with fractal statistics is explained by analogy to Browian motion (the archetypical fractal function): just as when a dust mote randomly bombarded by air molecules produces a fractal Brownian random walk, the complex interaction of processes that locally modify shape produces a fractal Brownian surface.

Naturally occuring fractal-like surfaces have two important properties: (1) each segment is statistically similar to all others; (2) segments at different scales are are statistically indistinguishable, i.e., as we examine such a surface at greater or lesser imaging resolution its statistics (curvature, etc.) remain the same. Because of these invariances, the most important *variable* in the description of such a shape is how it varies with scale; in essence, how many large features there are relative to the number of middle-sized and smaller-sized features. For fractal shapes (and thus for many real shapes) the ratio of the number of features of one size to the number of features of the next larger size is a constant — a surprising fact that derives from the property of scale invariance. The fractal model, therefore, leads us to characterize a surface's statistics in terms of two parameters: the surfaces variance (amplitude), and the ratio between the frequency of smaller and larger features (i.e., its fractal dimension).

Although quite useful in describing natural surfaces, this statistical fractal-based model has a very serious limitation: it does *not* describe the patterning of surfaces, only how their overall statistics vary with scale. We may remedy this restriction by use of the descriptive language outlined above. It turns out that fractal surfaces may be constructed by the Boolean combination of our "lumps," specifically, the recursive sum of smaller and smaller lumps, when carried to the limit, forms a true fractal surface. This construction is illustrated in Figure 7(a). We first pick a ratio r, $0 \leq r \leq 1$, which determines the fractal dimension of the surface (i.e., the fractal dimension D of a surface is determined by $D = T + r$, where T is the topological dimension of the surface) and randomly place n^2 large bumps on a plane, giving the bumps a Gaussian distribution of altitude (with variance σ^2), as seen in Figure 7(a). We then add to that $4n^2$ bumps of half the size, and altitude variance $\sigma^2 r^2$, as shown in Figure 7(b). We continue with $16n^2$ bumps of one quarter the size, and altitude $\sigma^2 r^4$,

[7] The defining characteristic of a fractal is that it has a *fractional dimension*, from which we get the word "fractal."

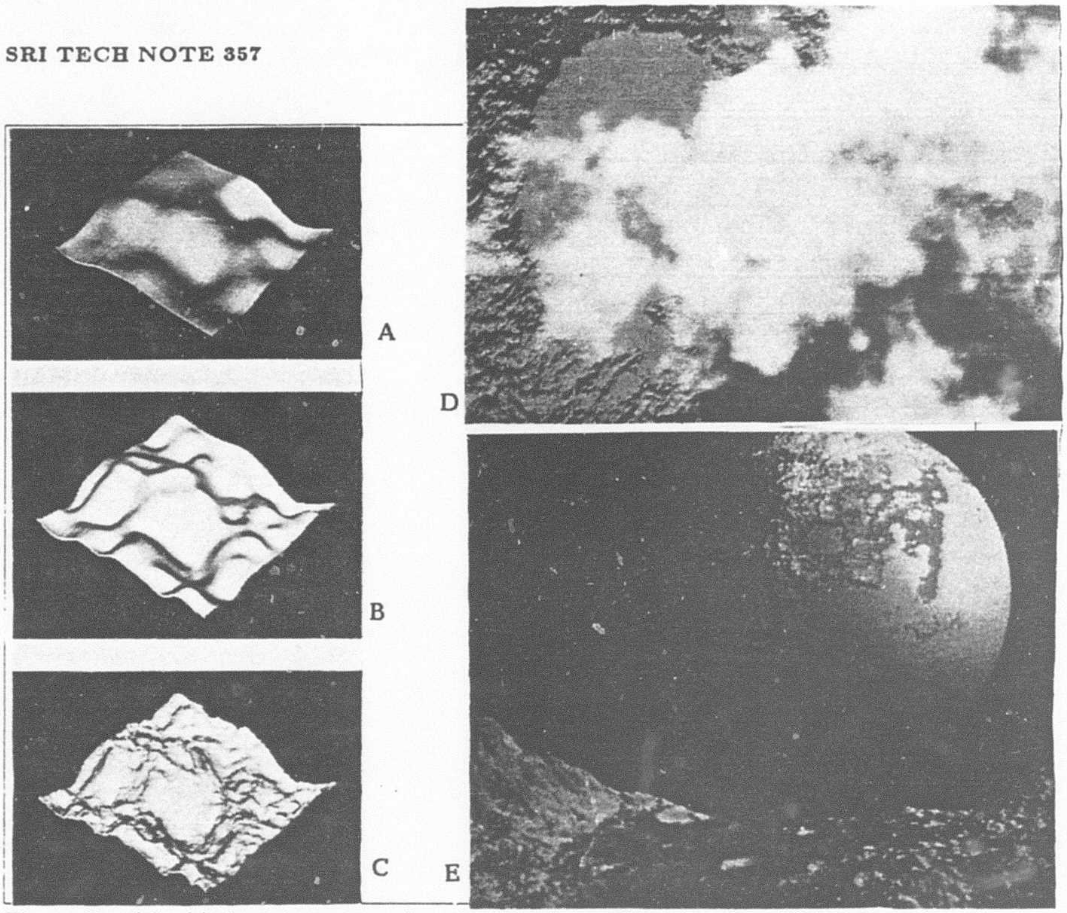


Figure 7. (a) - (c) show the construction of a fractal shape by successive addition of smaller and smaller features with number of features and amplitudes described by the ratio $1/r$. All of the forms and surfaces shown in (d) and (e) (which are images by Voss and Mandelbrot, see [22]) can be generated in this manner.

then $64n^2$ bumps one eighth size, and altitude $\sigma^2 r^6$ and so forth, as shown in Figure 7(c). The final result, shown in Figure 7(c) is a true Brownian fractal shape. This construction does not depend on the particular shape of the bumps employed;[8] the only constraint is that the sum must fill out the Fourier domain. Figures 7(d) and 7(e) illustrate the power and generality of this construction; all of the forms and surfaces in these images can be constructed in this manner.

When the placement and size of these lumps is random, we obtain the classical Brownian fractal surface that has been the subject of our previous research. When the larger components of this sum are matched to a particular object, however, we obtain a description of that object that is exact to the level of detail encompassed by the specified components. This makes it possible to specify a global shape while retaining a qualitative, statistical description at smaller scales: to describe a complex natural form such as a cloud or mountain, we specify the "lumps" down to the desired level of detail by fixing the larger elements of this sum, and then we specify only the fractal statistics of the smaller lumps thus fixing the qualitative appearance of the surface. Figure 8 illustrates an example of such

[8] Different shaped bumps will, however, give different appearance or texture to the resulting fractal surface; this is an important and as yet relatively uninvestigated aspect of the fractal model.

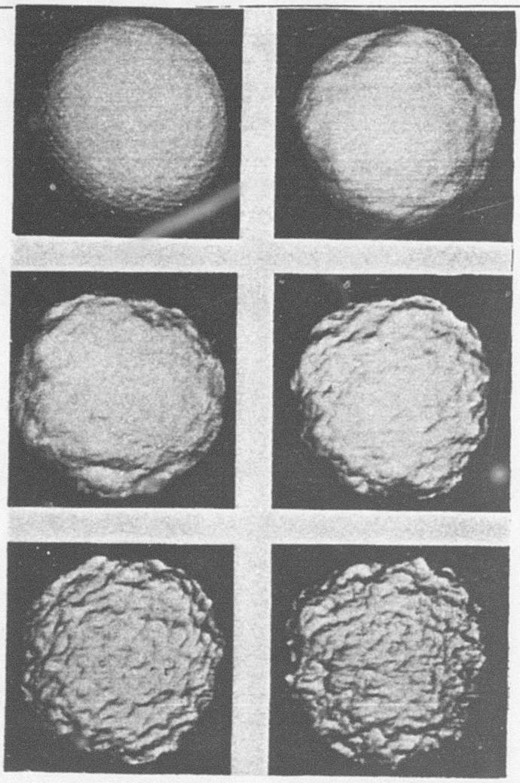


Figure 8. Spherical shapes with surface crenulations ranging from smooth (fractal dimension = topological dimension, $r \approx 0$) to rough (fractal dimension $>>$ topological dimension, $r \approx 1$).

description. The overall shape is that of a sphere; to this specified large-scale shape, smaller lumps were added randomly. The smaller lumps were added with six different choices of r (i.e., six different choices of fractal statistics) resulting in six qualitatively different surfaces — each with the same basic spherical shape.

The ratio r between the number of features of one size to the number of features at another size describes how the surface varies across different scales (resolutions, spatial frequency channels, etc.). This ratio, which is linearly related to the surfaces' fractal dimension, summarizes how complex the surface is; how many features of one size there are for each larger feature. It is an intrinsic property of the surface[9] ; surfaces formed by different processes typically have different a ratio/fractal dimension. Thus the ratio allows us to crudely classify the surface in terms of the process that formed it.[10] We have found that by measuring the fractal dimension (and thus the ratio r) in patches of the image we can infer the fractal dimension (and ratio r) for a homogeneous 3-D surfaces [34]; in experiments

[9] This ratio primarily depends on the spatial autocorrelation of the process that formed the surface

[10] Because formative processes tend to act over a range of scales, real surfaces normally have a constant ratio over fairly wide (e.g., 1 : 8) ranges of scale, although it is rare for the ratio to be constant over several decades of scale. Thus if we observe that the ratio at large scales is much different than at small scales (as in Figure 8), we can reliably infer that two different process were involved in forming the surface, and that they acted over different ranges of scale.

this has allowed us to closely predict people's perception of surface roughness [38,39]; we can speculate, therefore, that the demonstrated ability of people to preattentively segment an image on the basis of this ratio gives them a method of segmenting the scene into regions that were separately formed.

4 Primitive Perception: Recognizing Instances of Models

During the last decade, the dominant view of human perception has been that perception proceeds through successive levels of increasingly sophisticated representations until finally, at some point, information is transferred to our general cognitive faculties. And indeed, there *does* seem to be a gradient of sophistication in human perception, ranging from seemingly primitive inferences of shapes, textures, colors, and the like, to the apparently more sophisticated inferences of chairs, trees, affordances[11] and people's emotions. There is significant reason to believe, however, that this is not simply the flow of information through successive levels of representation.

To summarize Fodor's excellent and extended argument for this conclusion [40], we note that the sophisticated end of perception can involve virtually anything we know, and seems to blend smoothly into general cognition — for instance, we speak of perceiving abstract mathematical relationships or people's intentions. There is no principled reason to separate sophisticated perception from general purpose reasoning. The characteristics of primitive perception, however, are quite different from that of cognition:

- *Informational encapsulation:* Primitive perception proceeds without benefit of intimate access to the full range of our world knowledge. Most visual illusions, for instance, cannot be dispelled merely by recognizing them as illusions [41].
- *Limited extent:* The body of knowledge on which primitive perception draws is of quite limited extent, at least in comparison to our conscious world knowledge. People of all cultures seem to share a common perceptual framework [43]; it is this shared framework that makes possible any communication at all.
- *Functional autonomy:* Primitive perception proceeds with little regard to the particulars of the task at hand, under at most limited voluntary control. We are capable of the same discriminations, regardless of purpose or task, except (perhaps) for a few very practiced tasks, e.g., birdwatchers discriminating between different types of bird. This is not to say that we always *do* make the same discriminations (we can, after all, focus our attention), but rather that whenever we attend to a particular stimulus dimension we are always capable of making the same discriminations.

Primitive perception is at least roughly the realm of perceptual organization, i.e., the pre-attentive organization of sensory data into primitives like texture, color and form. Thus, although we often speak as if perception were a smooth series of progressively more sophisticated inferences [10], it is more likely that there are separate, specialized mechanisms for primitive and sophisticated inferences.

This leads to a conception of our perceptual apparatus as containing two distinct parts: the first, a special-purpose, perhaps innate mechanism that supports primitive perception,

[11] affordances are the purpose(s) of an object.

and the second something that clc-ely resembles general cognition. Most of the time the sensory data is first examined by the mechanisms of primitive perception to discover instances of rigidity, parallelism, part-like groupings and other evidences of causal organization, and then the mechanisms of sophisticated perception use specific, learned knowledge about the world to refine this primitive causal explaination into a detailed account of the environment.

It should be noted, however, that for at least the most practiced discriminations things seem to happen somewhat differently. When a percept, even if of a very sophisticated nature, is highly practiced or very important it appears that our minds build up a special-purpose mechanism solely for that purpose. Consider, for instance, peoples incredible facility at recognizing their own name, or the faces of familiar people. There may be, therefore, a sort of "compiler" for building specialized routines for these oft-repeated, important or time-critical discriminations. How much of our day-to-day perception is handled by such special-purpose routines is very much an open question.

Primitive perception, by our definition, was first seriously addressed by the Gestalt psychologists [4], who noticed that people seem to spontaneously impose a physically meaningful organization upon visual stimuli, through grouping, figure/ground separation, and so forth. They found that the addition of semantic context very rarely affects this spontaneous, pre-attentive organization of the image; somehow the visual system seems able to group an image into the correct, physically meaningful parts *before* contextual knowledge is available.

The Gestalt psychologists described this spontaneous organization as being governed by the principle of *Pragnanz*[12] , however their lack of the modern notions of computation limited their ability to crisply define *Pragnanz* and thus doomed them to a rather limited success. Nevertheless, their work paved the way for the two-stage model of perception that is enjoying widespread popularity in academic circles today. The first stage, which we are describing here as primitive perception, is spontaneous and pre-attentive. It carves the sensory data into likely-meaningful parts, and presents them to the later stages of perception. The second stage of perception, which we are calling sophisticated perception, is very little (if at all) different from our general cognitive faculty — including the ability to make very efficient, "compiled" routines, presumably by combining the outputs of primitive perception.

4.1 Recognizing Our Modeling Primitives

It is our goal to provide the beginnings of a theory for our faculty of pre-attentive, primitive perception: to present a rigorous, mathematical definition for the vague notion of "a part" and to explain how we can, Gestalt-like, carve an image up into meaningful "parts" without need of semantic context or specific a priori knowledge. We have already described a representation that is competent to describe a wide range of natural forms, and whose primitive elements seem to correspond closely to our naive notions of perceptual parts. What remains is to be done is to show that these descriptive primitives can be recovered from the image data.

The major difficulty in recovering such descriptions is that image data is mostly a function of surface normals, and not directly a function of the surface shape. This is

[12] *Pragnanz* is normally translated as meaning "goodness of form"

because image intensity, texture anisotropy, contour shape, and the like — the information we have about surface shape — is largely determined by the direction of the surface normal. To recover the shape of a general volumetric primitive, therefore, we must (typically) first compute a dense depth map from information about the surface normals. The computation of such a depth map has been the major focus of effort in vision research over the last decade and, although the final results are not in, the betting is that such depth maps are impossible to obtain in the general, unconstrained situation. Even given such a depth map, the recovery of a shape description has proven extremely difficult, because the parameterization of the surface given in the depth map is generally unrelated to that of the desired description.

Because image information is largely a function of the surface normal, one of the most important properties of superquadrics is the simple "dual" relation between their surface normal and their surface shape. It appears that this dual relationship can allow us to form an overconstrained estimate of the 3-D parameters of such a shape from noisy or partial image data, as outlined by the following equations.

The surface position vector of a superquadric with length, width and breadth a_1, a_2 and a_3 is (again writing $\cos\eta = C_\eta$, $\sin\omega = S_\omega$)

$$\mathbf{X}(\eta,\omega) = \begin{pmatrix} a_1 C_\eta^{\epsilon_1} C_\omega^{\epsilon_2} \\ a_2 C_\eta^{\epsilon_1} S_\omega^{\epsilon_2} \\ a_3 S_\eta^{\epsilon_1} \end{pmatrix} \tag{1}$$

and the surface normal at that point is

$$\mathbf{N}(\eta,\omega) = \begin{pmatrix} \frac{1}{a_1} C_\eta^{2-\epsilon_1} C_\omega^{2-\epsilon_2} \\ \frac{1}{a_2} C_\eta^{2-\epsilon_1} S_\omega^{2-\epsilon_2} \\ \frac{1}{a_3} S_\eta^{2-\epsilon_1} \end{pmatrix} \tag{2}$$

Therefore the surface vector $\mathbf{X} = (x, y, z)$ is dual to the surface normal vector $\mathbf{N} = (x_n, y_n, z_n)$ in the following sense:

$$\mathbf{N}(\eta,\omega) = \begin{pmatrix} \frac{1}{x} C_\eta^2 C_\omega^2 \\ \frac{1}{y} C_\eta^2 S_\omega^2 \\ \frac{1}{z} S_\eta^2 \end{pmatrix} \tag{3}$$

From (1) and (3), then we have

$$x_n = \frac{C_\eta^2 C_\omega^2}{x} \qquad y_n = \frac{C_\eta^2 S_\omega^2}{y} \tag{5}$$

so

$$\frac{y_n}{x_n} = \frac{x}{y} \tan^2\omega \tag{6}$$

or

$$\left(\frac{y y_n}{x x_n}\right)^{1/2} = \tan\omega \tag{7}$$

We may also derive alternative expressions for $\tan\omega$ as follows:

$$x = a_1 C_\eta^{\epsilon_1} C_\omega^{\epsilon_2} \qquad y = a_2 C_\eta^{\epsilon_1} S_\omega^{\epsilon_2} \tag{8}$$

so

$$\frac{x}{y} = \frac{a_1}{a_2}\left(\frac{C_\omega}{S_\omega}\right)^{\epsilon_2} \tag{9}$$

or

$$\left(\frac{ya_1}{xa_2}\right)^{1/\epsilon_2} = \tan\omega \tag{10}$$

Combining these expressions for $\tan\omega$ we obtain

$$\left(\frac{yy_n}{xx_n}\right)^{1/2} = \left(\frac{ya_1}{xa_2}\right)^{1/\epsilon_2} \tag{11}$$

or

$$\frac{y_n}{x_n} = (\frac{y}{x})^{2/\epsilon_2 - 1}\left(\frac{a_1}{a_2}\right)^{2/\epsilon_2} \tag{12}$$

Letting $r = y_n/x_n$, $k = (a_1/a_2)^{2/\epsilon_2}$ and $\xi = 2/\epsilon_2 - 1$ we find that

$$r = k(\frac{y}{x})^\xi \tag{13}$$

$$\frac{dr}{dy} = \frac{k\xi}{x}(\frac{y}{x})^{\xi-1} \tag{14}$$

$$\frac{dr}{dx} = \frac{-k\xi y}{x^2}(\frac{y}{x})^{\xi-1} \tag{15}$$

This gives us two equations relating the unknown shape parameters to image measureable quantities, i.e.,

$$\frac{r}{\frac{dr}{dy}} = \frac{y}{\xi} \tag{16}$$

and

$$\frac{r}{\frac{dr}{dx}} = \frac{-x}{\xi} \tag{17}$$

Thus Equations (16) and (17) allow us to construct a linear regression to solve for center and orientation of the form, as well as the shape parameter ϵ_2, given only that we can estimate the surface tilt direction r.

Overconstraint and reliablity. Perhaps the most important aspect of these equations is that we can form an *overconstrained* estimate of the 3-D parameters: thus we can *check* that our model applies to the situation at hand, and we can *check* that the parameters we estimate are correct. This property of overconstraint comes from using models: when we have used some points on a surface to estimate 3-D parameters, we can check if we are correct by examining additional points. The model predicts what these new points should look like; if they match the predictions then we can be sure that the model applies and that the parameters are correctly estimated. If the predictions do *not* match the new data points, then we know that something is wrong. The ability to check your answer is perhaps the most important property any vision system can have, because only when you can check your answers can you build a reliable vision system. And it is *only* when you have a model that relates many different image points (such as a model of how rigid motion appears in an image sequence, or a CAD-CAM model, or this 3-D shape model) that you can have the overconstraint needed to check your answer.

One other aspect of Equations (16) and (17) that deserves special note is that the only image measurement needed to recover 3-D shape is the surface tilt τ, the component of shape that is unaffected by projection and, thus, is the most reliably estimated parameter of surface shape. It is, for instance, known exactly at smooth occluding contours and both shape-from-shading and shape-from-texture methods produce a more reliable estimate of τ than of slant, the other surface shape parameter. That we need only the (relatively) easily estimated tilt to estimate the 3-D shape parameters makes robust recovery of 3-D shape much more likely.

When we generalize these equations to include unknown orientation and position parameters for the superquadric shape, we obtain a new set of nonlinear equations that can then be solved (in closed form) for the unknown shape parameters ϵ_1 and ϵ_2, the center position, and the three angles giving the objects orientation. Once these unknowns are obtained the remaining unknowns (a_1, a_2, and a_3, the three dimensions of the object) are easily obtained.

For the case of rotation and translation in the image plane, the equations become:

$$x^* = C_\theta(x - x_0) + S_\theta(y - y_0) \qquad y^* = -S_\theta(x - x_0) + C_\theta(y - y_0) \tag{18}$$

where θ is the rotation, x_0, y_0 the translation, and (x^*, y^*) the new rotated and translated coordinate system. The tilt τ then becomes

$$\tau = \frac{y_n^*}{x_n^*} = \frac{(-S_\theta x_n + C_\theta y_n)}{(C_\theta x_n + S_\theta y_n)} \tag{19}$$

and the derivative of Equation (19) is

$$\begin{aligned}\frac{d\tau}{dy^*} &= (-S_\theta \frac{dx_n}{dy^*} + C_\theta \frac{dy_n}{dy^*})(C_\theta x_n + S_\theta y_n)^{-1} \\ &\quad - (-S_\theta x_n + C_\theta y_n)(C_\theta x_n + S_\theta y_n)^{-2}(C_\theta \frac{dx_n}{dy^*} + S_\theta \frac{dy_n}{dy^*}) \\ &= (C_\theta x_n + S_\theta y_n)^{-2}\left(x_n \frac{dy_n}{dy^*} - y_n \frac{dx_n}{dy^*}\right)\end{aligned} \tag{20}$$

Noting that

$$\frac{dx_n}{dy^*} = \frac{dx_n}{dy}\frac{dy}{dy^*} = \frac{dx_n}{dy}C_\theta^{-1} \qquad \frac{dy_n}{dy^*} = \frac{dy_n}{dy}\frac{dy}{dy^*} = \frac{dy_n}{dy}C_\theta^{-1} \tag{21}$$

Equation (16) can now be rewritten as

$$\begin{aligned}&(C_\theta x_n + S_\theta y_n)(-S_\theta x_n + C_\theta y_n) = \\ &\qquad \frac{1}{C_\theta \xi}[-S_\theta(x - x_0) + C_\theta(y - y_0)]\left(x_n \frac{dy_n}{dy} - y_n \frac{dx_n}{dy}\right)\end{aligned} \tag{22}$$

Our estimates of tilt from local image information typically have considerable noise in them [18,37,44]; in order to still obtain a good estimate of three-dimensional shape we will formulate the problem of recovering the shape parameters as a linear regression. Collecting the image-measureable terms together (in square brackets), this equation becomes

$$\begin{aligned}0 = \ & [x_n^2 - y_n^2](\xi C_\theta^2 S_\theta) \\ & + [x_n y_n](\xi C_\theta(S_\theta^2 - C_\theta^2)) \\ & + [x_n \frac{dy_n}{dy} - y_n \frac{dx_n}{dy}](S_\theta x_0 - C_\theta y_0) \\ & + [x x_n \frac{dy_n}{dy} - x y_n \frac{dx_n}{dy}](-S_\theta) \\ & + [y x_n \frac{dy_n}{dy} - y y_n \frac{dx_n}{dy}](C_\theta)\end{aligned} \tag{23}$$

This equation, then, can be used for a linear regression to solve for the unknown coefficients (in curved brackets). We have five unknown coefficients and so we require tilt information at as few as five points in order to solve for all these unknowns; from these we can obtain closed form solutions for the center of the object (x_0,y_0), the shape parameter ϵ, and the orientation θ. In fact, things are somewhat better than this, because we have *two* such equations at each point (one for dx and one for dy) so that fewer points are actually required. The small number of points required opens up the possibility of segmenting images in terms of the parameters of the 3-D surface.

At occluding contours the situation is better yet, because we also know that $y_n^2 + x_n^2 = 1$, and considerable extra constraint is available. This formulation, therefore, reflects the fact that contour information is more powerful than shading or texture information. One of the more interesting aspects of this approach is that contour information and information from shading or texture contribute toward estimating shape in exactly the same manner — by providing information about surface tilt — and therefore we may combine information from all of these sources by use of the same set of equations, those derived from Equations (16) and (17).

Because we have formulated the problem of primitive perception as one of recognizing instances of the "parts" found in a representational vocabulary, we can frame the problem as one in statistical decision theory: we have a range of hypotheses that we entertain, and use image data to decide among the alternatives. This gives us a rigorous framework for integrating information from motion, stereo, etc., together with contour, shading and texture information without having to make further assumptions. This is in considerable contrast to approaches that try to apply strong, unverifiable assumptions about the nature of surfaces (e.g., that all surfaces are "smooth") in order to integrate various information sources. Here we are attempting to collect a vocabulary of models that span the space of shape possiblities, so that we can replace unverifiable assumptions with verifiable models. We want perception to proceed by making an overconstrained, statistical determination that a particular model is applicable (rather than simply making an assumtion), and then estimate the parameters of that model. If our vocabulary of shape does in fact cover the range of shape that actually occurs, then *we will have made the best shape estimate possible with the available image data.*

Equation (23) does not reflect the full sophistication possible in statistical decision theory; a regression using this equation results in a maximum likelihood estimate of compound parameters such as C_θ and $\xi C_\theta^2 S_\theta$ rather than estimates of the individual parameters ϵ and θ. Still, the main power of the approach remains. Our modeling primitives provide us with a parameterized range of hypotheses that we can choose among using established statistical tools, thus providing us a rigorous framework for integrating contour with shading and texture information, as well as allowing us to include *a priori* information that we may have gained from previous views. The power of this framework has been illustrated by the work of Ferrie and Levine [44] who, using a simpler shape vocabulary consisting of ellipsoids and cylinders, have combined our local shape-from-shading technique [18] with motion information to accurately recover 3-D shape.

Although the equations presented here are only for rotations in the image plane, the general equations are similar, although somewhat more complicated. As in the simpler case, information at relatively few points[13] is required in order to solve for the unknowns, and the situation is considerably better along occluding contours.

Figure 9 illustrates the process of recovering 3-D shape using this technique. Figure 9(a) shows a half-toned version of an image of a superquadric with shape parameters $e_1, e_2 = 0.5$. To this image, we applied the local shape-from-shading/texture technique developed by Pentland [18,37]. The estimation technique employs second-derivative filters with local support to make estimates of surface slant and tilt, with the estimates of tilt being more reliable than the estimates of slant [18,44]. Figure 9(b) shows a view of the surface tilt (i.e., y_n/x_n) recovered from the continuous 8-bit image of the shape illustrated by Figure 9(a); in this figure the image x axis runs left-right and the y axis runs up-down. From this estimated tilt surface we can use Equations 16 and 17 to estimate the center of the shape, the shape parameter e_2, and the width and breadth of the shape. Figure 9(c) shows two views of the recovered shape; it can be seen that in this simple case a good estimate of the 3-D shape can be made. It appears, then, that Equations (16) and (17) offer a good hope for recovering surface shape; in our future research we hope to extend these results to natural imagery.

4.2 Model-Based Vision, the Blocks' World, and Our Effort

The most successful (i.e., working, practical) efforts in machine vision have all been accomplished within two paragdigms that are generally lumped together under the rubric of "model-based vision." The first of these paradigms is to take a CAD-CAM type model of a specific object, find configurations of image features that uniquely determine identity and orientation of the object, and then search the image for those configurations. A similar, but fundamentally quite different "model-based vision" paradigm was first employed in the Blocks' world research during the 1960's [see Roberts (45)], and more recently in such work as the 3-D Mosaic program of Hermann and Kanade [24]. In this second paradigm, the

[13] Depending on the exact formulation, 15 points are required.

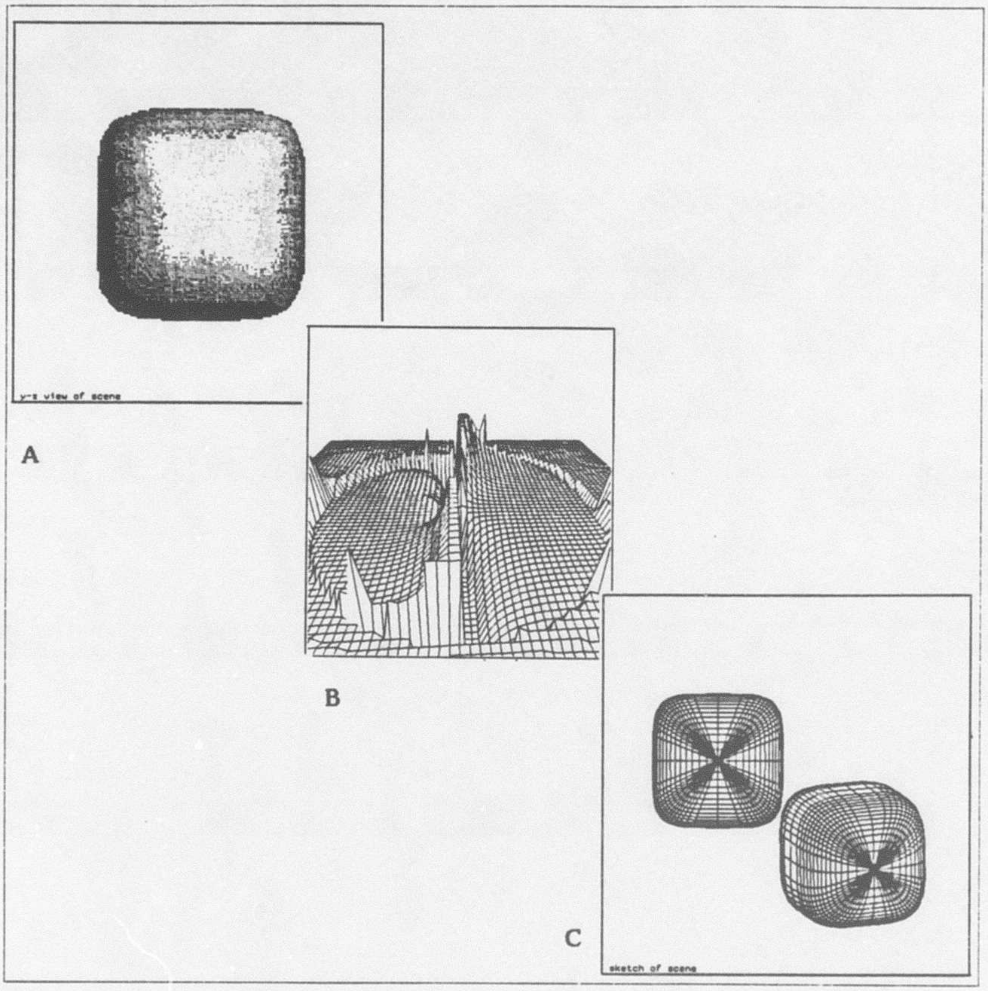


Figure 9. (a) A half-toned version of an image of a superquadric with shape parameters $e_1, e_2 = 0.5$, (b) the surface tilts estimated using the local shape-from-shading/texture algorithm described in Appendix B, (c) two views of the 3-D shape estimated by use of equations 16 and 17, using the tilt estimates shown in (b).

models are of the *parts* that make up specific objects, rather than a models of the entire object, and the goal is to identify those component parts. Once the parts have been identified and their spatial layout determined, one can ask if this configuration of parts is an object that has been seen before. This latter approach has the very significant advantage that it can learn new object descriptions by example: it can look at a new object, identify the object's parts, and then use that part-wise description to build up a general model of the specific object in a manner similar to that proposed by Winston [46,47].

Because this second, find-the-parts approach to model-based vision can learn descriptions of novel objects, it has the potential to support general-purpose vision. The major limitation on the success of this approach is the availablity of part models that are individually recognizable and which have the expressive power to describe everything within the domain of interest. What we are attempting to do is develop a vocabulary of just such individually recognizable part models. One may, therefore, think of the research described here as returning to the Blocks world, but with models of 3-D structure that are tremendously more sophisticated than simple blocks or polyhedra.

We believe that the modeling language presented here has a good chance of being able to handle most of the forms found in the real world. The images in this paper demonstrate the expressive power of this new vocabulary of models (their cartoon-like nature is primarily due to the lack of surface texturing), and the mathematics in this section of the paper demonstrate the plausablity of recovering such part descriptions from sparse and partial image data. Even if it should turn out that our models aren't yet sophisicated enough to deal with the complexity of real world, we will have *at least* made major progress towards bridging the gap between the present state of the art and that needed to construct a general-purpose, real-world vision system.

5 Using The Representation

The particular models of the world that perception uses to interpret sensory data induce a profound organization on all of our conceptual structures. If we stand in the center of Stonehedge, we can see either a collection of pillars, several irregular walls of pillars, or concentric circular structures with regularly-spaced pillars. This is the familar Gestalt phenominon of grouping; what is important about it is that *which* grouping you spontaneously see strongly influences what hypotheses you entertain when trying to deduce, for instance, the purpose of Stonehedge. Examples such as this demonstrate that the manner in which perception "carves up" the world — that is, its models of the world — strongly determine the way in which we think about the world.

The issue of perceptual models is, therefore, of more than passing interest to those interested in cognition. It seems reasonable that if we are to develop machines that are able to display commonsense reasoning abilities, for instance, we must have spatial representations that are at least roughly equivalent to those people employ in organizing their picture of the world. Similarly, if we are ever to communicate with machines about our shared environment we must develop spatial representations that are at least isomorphic to the representations that we use. We must have a representation that captures the same sorts of distinctions we make when we carve objects into parts.

Because communication depends upon having a shared representation of the situation, we can use man-machine communication as a fairly sensitive test of whether a particular representation captures the notions of difference and similarity that humans employ. The empirical (and so far informal) finding that the organization of our shape descriptions correspond closely with the human perceptual organization is, as a consequence, quite interesting: the representation seems to offer exciting possiblities for flexible, effective man-machine communication. It was therefore of great interest to test how effectively we can use the representation described here as a basis for communication between a computer and its operator concerning image data and 3-D shape.

5.1 Communicating about a Digitial Terrain Map

As a first experiment we took the problem of communicating with a computer about a digital terrain map, as might be done in guiding a stereo compilation process or when plotting a path through the terrain. Figure 7 showed how a mountainlike surface can be built up from the combination of progressively smaller primitives. We can also take a real surface, such as the digital terrain map of Yosemite Valley shown in Figure 10(a) and decompose it into a canonical lump-description by use of a minimum-complexity criterion, that is, we attempt to account for the shape with the fewest number of component parts as is possible (see Szeliski [49]). One simple mechanism for approximating this decomposition is to form a Laplacian pyramid [50], examine the entries in this pyramid to find those points that most closely correspond to the shape of a single "lump"(by looking at the neighbors of the point in both space and scale), subtract off that lump from the original form, and repeating this procedure until no entries remain in the pyramid.

If we want to have a "sketch" of the DTM surface (a simplified description that we can use for communication), we can use estimates of the surface's variance and fractal dimension to set an acceptance threshold, so that our decomposition proceedure finishes by taking only the 50 or so most prominent surface features. To adequately characterize a DTM we have found that we need to look for only two types of primitive elements: one, a vertically-oriented symmetrical peak, and two, a horizontally-oriented elongated ridge or valley. The fractal statistics of the surface characterize how features of the surface change across scales and therefore gives us the information needed to adjust the acceptance threshold for different scales, so that the prominence of features accepted at one scale corresponds to the prominence of smaller or larger scale features. When we do this, the result is a description that organizes the pixel data into its most prominent components at all scales, in a way that we have found corresponds closely with our naive perceptual organization of the surface — e.g., organizing the surface into peaks, ridges, valleys, and the like.

The ability to structure the pixel data in a manner that corresponds to the perceptual organization we impose upon the data allows us to support human-computer communication about the scene. It allows us to point to a part of the scene, say "that thing" and have the machine be able to make a good guess about what part of the surface we want to indicate, as opposed to the current state-of-the-art in which we have to carefully outline the part of the surface that we want manipulate.

This sort of communication is illustrated in Figures 10(b), (c) and (d), which shows the operation of a program we have constructed that performs this parsing of a Digital Terrain Map (DTM), identifies the 50 or so most prominent perceptual "parts," and then allows the user to interact with the DTM by simply pointing to peaks, valleys, ridges and so forth. These figures show a user pointing, the program interpreting what "feature" the user intended to indicate, and then highlighting that feature by cross-hatching it. The highlighted feature can then be edited to improve the DTM, defined as a primitive object in a path planning calculation, or used in whatever manner the user's purpose demands. As these figures illustrate, we have found a good correspondence between this program's structuring of the image and the structure people impose on the image.

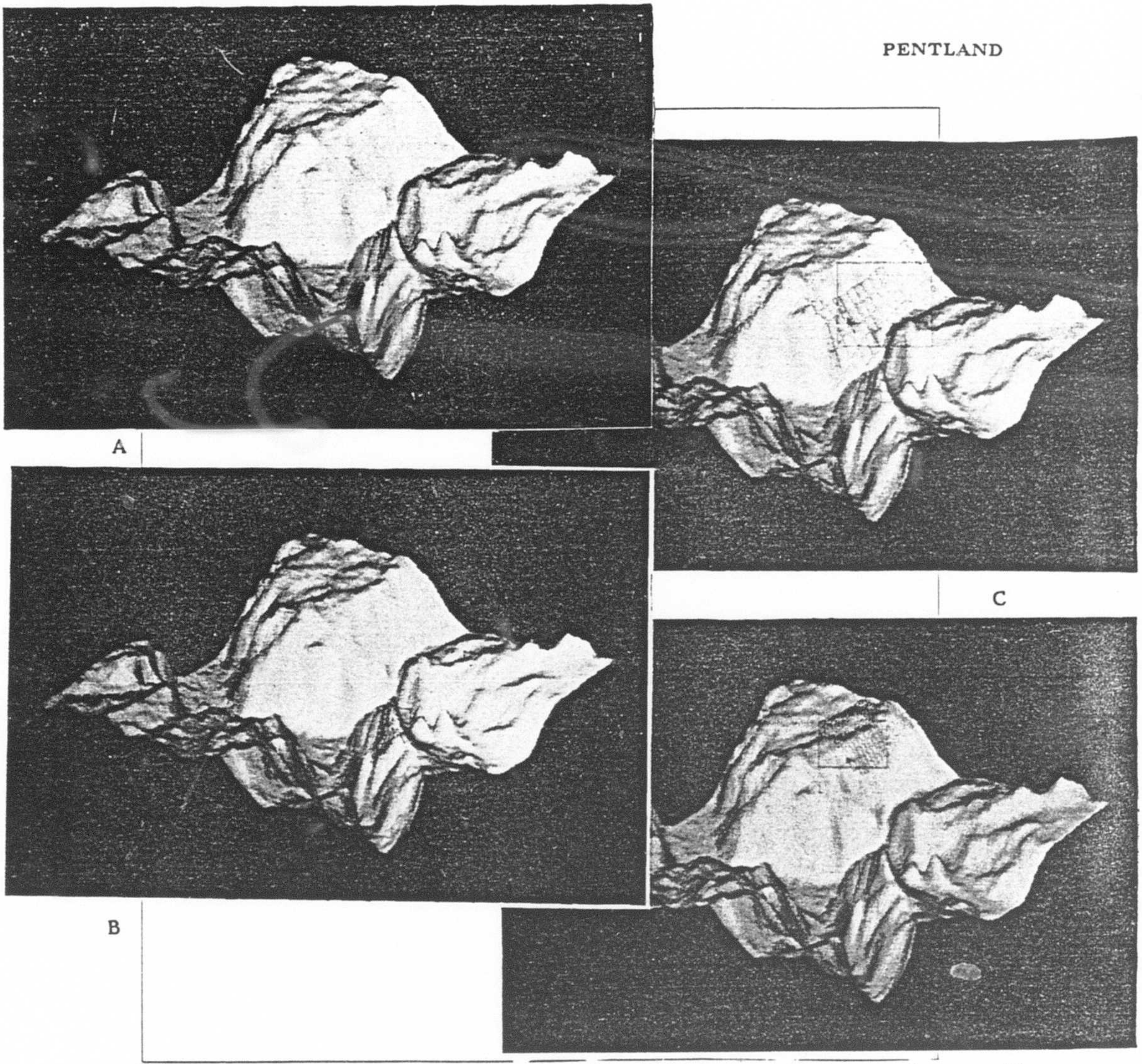


Figure 10. (a) a digital terrain map of Yosemite Valley, which is automatically decomposed into a "sketch," a description in our representational system that contains terms ("lumps") that correspond roughly to "peaks" "valleys" and "ridges," so that the parts of this description correspond closely with the perceptual organization that we impose on the scene. This is illustrated in (b), (c) and (e), which show a person pointing to a part of the image, and the computer using this sketch to determine what part of the terrain is being gestured at, and highlighting the "part" referred to by covering it with crosshatching. This decomposition of the scene into perceptually salient "parts" thus fulfills a critical requirement for effective man-machine communication: similar representations of the scene.

5.2 Building 3-D models

One other example that illustrates using the representation to facilitate man-machine communication is the 3-D modeling system called "SuperSketch" that was used to make most of the images in this paper. In this Symbolics 3600-based modeling system users create "lumps," change their squareness/roundness, stretch, bend, and taper them, and make Boolean combinations of them in real time by moving the mouse through the relevent parameter space, controlling which parameter is being varied by using the mouse buttons. Because these forms have an underlying analytical form, we can use fast, qualitative approximations to accomplish hidden surface removal, intersection and image intensity calculations in real time — something that could not be accomplished on a 3600 if a polygon-based description were employed. "Real time" in this case means that an "lump" can be moved, hidden surface removal accomplished, and drawn as a 100 polygon line drawing approximation in 1/8th of a second, and a complex, full color image such as Figure 1 can be rendered in approximately 20 seconds[14] .

Because the primitives, operations and combining rules used by the computer are very well matched to those of the human operator, we have found that interaction is suprisingly effortless: it took a relatively unskilled operator less than a half-hour to assemble the face in Figure 6, about ten minutes to create the lobster in Figure 3, and about four hours total to make Figure 1. This is in rather stark contrast to more traditional 3-D modeling systems that might require days or weeks to build up a scene such as shown in Figure 1. This performance, perhaps more than any other statistic that could be given, illustrates how the close match between this representational system and the perceptual organization employed by human operators facilitates effective man-machine communication.

6 Summary

To support our reasoning abilities perception must recover environmental regularities — e.g., rigidity, "objectness", axes of symmetry — for later use in cognitive processes. Understanding this recovery of structure is critically important because the structural organization that perception delivers to cognition is the foundation upon which we construct our picture of the world; these regularities are the building blocks of all cognitive activities.

To understand how our perceptual apparatus can produce meaningful cognitive building blocks from the unstructured array of image intensities we would like to have a representation that correctly models both important environmental regularities and also accounts for the perceptual organization we impose on the stimulus — the one structuring of the stimulus that we know can support general-purpose cognitive activity. Unfortunately, the representations that are currently available were originally developed for other purposes (e.g., the point-wise descriptions of physics, or the platonic-solids descriptions of engineering) and therefore are often unsuitable for the problems of perception or commonsense reasoning.

For instance, the complexity of standard descriptions for such common natural forms as clouds, human faces, or trees is a fundamental block to progress in artificial intelligence

[14] A Symbolics 3600 is approximately the speed of a VAX 11/780, except for floating point operations (used extensively in SuperSketch) which are almost an order of magnitude slower.

and machine vision: How can we hope to recover 3-D shape descriptions from an image when the number of parameters to be recovered approximately equals the number of pixels in the image? How can we hope to reason about such an overly complex description effectively?

In answer to these problems we have presented a representation that has proven competent to accurately describe an extensive variety of natural forms (e.g., people, mountains, clouds, trees), as well as man-made forms, in a succinct and natural manner. The approach taken in this representational system is to describe scene structure at a scale that is more like our naive perceptual notion of "a part" than the point-wise descriptions typical of current image understanding research, and to use a description that reflects a possible formative history of the object; e.g., how the object might have been constructed from lumps of clay.

Each of the component parts of this representation — superquadric "lumps," deformations, Boolean combination, and the recursive fractal construction — have been previously suggested as elements of various shape descriptions, usually for other purposes. The contribution of this paper is to bring all of these separate descriptive elements together, and employ them as a representation for natural forms and as a theory of perceptual organization. In particular, we believe that the important contributions of this paper are the following.

- We have demonstrated that this process-oriented representational system is able to accurately describe a very wide range of natural and man-made forms in an extremely simple, and therefore useful, manner. Further, the representation can be used to support fast, qualitative approximations to determine, e.g., intersection, appearance or relative position. Such qualitative reasoning is employed in SuperSketch allow real-time movement, deformation, Boolean combination, hidden surface removal, intersection and rendering.
- We have found that descriptions couched in this representation are similar to people's (naive) verbal descriptions and appear to match people's (naive) perceptual notion of "a part;" this correspondence is strong evidence that the descriptions we form will be good spatial primitives for the Naive Physics research program [11] and for commonsense reasoning in general. Additionally, we hope that this descriptive system will provide the beginning a rigorous, mathematical treatment of the still vaguely defined subject of human perceptual organization.
- The part-model approach to perception makes the problem of recovering shape descriptions overconstrained and therefore potentially extemely reliable, while still providing the flexibility to learn new object descriptions. One important goal of this paper, therefore, is to begin the process of replacing unverifiable *assumptions* with verifiable *models*. Toward this end we have shown that our current descriptive vocabulary is capable of describing a wide range of natural forms, and that the primitive elements of this language can be recovered from partial image data in an overconstrained and apparently noise-insensitive manner.
- And finally, we have shown that descriptions framed in the representation have markedly facilitated man-machine communication about both natural and man-made 3-D structures. It appears, therefore, that this representation gives us the right control knobs for discussing and manipulating 3-D forms.

The representational framework presented here is *not* complete. It seems clear that additional process-oriented modeling primitives, such as branching structures [21] or particle systems [51], will be required to accurately represent objects such as trees, hair, fire, or river rapids. Further, it seems clear that domain experts form descriptions differently than naive observers, reflecting their deeper understanding of the domain-specific formative processes and their more specific, limited purposes. Thus, accounting for expert descriptions will require additional, more specialized models. Nonetheless, we believe this descriptive system makes an important contribution toward solving current problems in perceiving and reasoning about natural forms, by allowing us to construct accurate models that are still simple enough to be useful, and by providing us with the basis for more effective man-machine communication.

Acknowledgement. I would like to thank Andy Witkin for his help and feedback in developing my ideas, and to thank David Heeger, Oscar Firschein, and Bob Bolles for their help and criticism in writing this manuscipt.

IV. REFERENCES

[1] Thompson, D'Arcy, (1942) *On Growth and Form,*, 2d Ed., Cambridge, England: The University Press.

[2] Stevens, Peter S., (1974) *Patterns In Nature,* Boston: Atlantic-Little, Brown Books.

[3] Rosch, E. (1973) On the internal structure of perceptual and semantic categories. In *Cognitive Development and the Acquisition of Language.* Moore, T.E. (Ed.) New York: Academic Press.

[4] Wertheimer, M. (1923) Laws of organization in perceptual forms, in *A Source Book of Gestalt Psychology,* W.D. Ellis (Ed.), New York: Harcourt Brace.

[5] Johansson, G., (1950) *Configurations in Event Perception,* Stockholm: Almqvist and Wiksell.

[6] Marr, D. and Nishihara, K., (1978) Representation and recognition of the spatial organization of three-dimensional shapes, *Proceedings of the Royal Society - London B,* 200:269-94

[7] Nishihara, H.K., (1981) Intensity, visible-surface and volumetric representations. *Artificial Intelligence 17,* 265-284.

[8] Binford, T. O., (1971) Visual perception by computer, *Proceeding of the IEEE Conference on Systems and Control,* Miami, December.

[9] Gibson, J. J., (1979) *The Ecological Approach to Visual Perception,* Boston: Houghton Mifflin.

[10] Marr, D. (1982) *Vision,* San Fransico: W.H. Freeman and Co.

[11] Agin, G.J., and Binford, T.O.,(1976) Computer descriptions of curved objects. *IEEE Transactions on Pattern Analysis and Machine Intelligence, C-25,* 4, 439-449.

[12] Nevatia, R., and Binford, T.O.,(1977) Description and recognition of curved objects. *Artificial Intelligence, 8,* 1, 77-98.

[13] Badler, N. and Bajacsy, R., (1978) Three-dimensional representations for computer graphics and computer vision, *Computer Graphics, 12,* 153-160.

[14] Brady, J. M., (1982) Describing visible surfaces. In *Computer Vision Systems,* A. Hanson and E. Riesman (Eds.)

[15] Brooks, R., (1985) Model based 3-D interpretation of 2-D images, In *From Pixels to Predicates,* Pentland, A. (Ed.) Norwood N.J.: Ablex Publishing Co.

[16] Bolles, B. and Haroud, R., (1985) 3DPO: An inspection system. In *From Pixels to Predicates,* Pentland, A. (Ed.) Norwood N.J.: Ablex Publishing Co.

[17] Barrow, H. G., and Tenenbaum, J. M., (1978) Recovering intrinsic scene characteristics from images. In *Computer Vision Systems,* Hanson, A. and Riseman, E. (Ed.) New York: Academic Press.

[18] Pentland, A. Local analysis of the image, (1984) *IEEE Transactions on Pattern Analysis and Machine Recognition, 6* 2, 170-187

[19] Witkin, A. P., and Tenenbaum, J. M., (1985) On perceptual organization. In *From Pixels to Predicates,* Pentland, A. (Ed.) Norwood N.J.: Ablex Publishing Co.

[20] A. Pentland and A. Witkin, (1984) "On Perceptual Organization," Second Conference on Perceptual Organization, Pajaro Dunes, CA, June 12-15.

[21] Smith, A. R., (1984) Plants, fractals and formal languages. In Computer Graphics 18,

No. 3, 1-11.

[22] Mandelbrot, B. B., (1982) *The Fractal Geometry of Nature*, San Francisco: Freeman.

[23] Georgeff, M.P., and Wallace, C.S., (1984) A general selection criterion for inductive inference. In Proceedings of the Sixth European Conference on Artificial Intelligence, 1984, Pisa, Italy, September 5-7.

[24] Herman, M. and Kanade, T. (1985) The 3-D mosaic scene understanding system, In *From Pixels to Predicates,* Pentland, A. (Ed.) Norwood, N.J.: Ablex Publishing Co.

[25] Konderink, Jan J., and van Doorn, Andrea J., (1982) The shape of smooth objects and the way contours end, *Perception, 11,* pp. 129-137

[26] Konderink, Jan J., and van Doorn, Andrea J., (1979) The internal representation of solid shape with respect to vision, *Biological Cybernetics, 32,* pp. 211-216

[27] Hoffman, D., and Richards, W., (1985) Parts of recognition, In *From Pixels to Predicates*, Pentland, A. (Ed.) Norwood, N.J.: Ablex Publishing Co.

[28] Barr, A., (1981) Superquadrics and angle-preserving transformations, *IEEE Computer Graphics and Application, 1* 1-20

[29] Kauth, R., Pentland, A. and Thomas, G. (1977) BLOB: an unsupervised clustering approach to spatial grouping, *Proceeding of the Eleventh International Symposium on Remote Sensing of the Environment,* Ann Arbor, Mich., April.

[30] Hobbs, J. (1985) Final Report on Commonsense Summer. SRI Artificial Intelligence Center Technical Note 370.

[31] Barr, A., (1984) Global and local deformations of solid primitives. *Computer Graphics 18,* 3, 21-30

[32] Hollerbach, J.M., (1975) Hierarchical shape description of objects by selection and modification of prototypes, M.I.T. Ph.D. Thesis, M.I.T. AI Technical Report 346.

[33] Hayes, P. (1985) The second naive physics manifesto, In *Formal Theories of the Commonsense World,* Hobbes, J. and Moore, R. (Ed.), Norwood, N.J.: Ablex

[34] Pentland, A. (1984a), Fractal-based description of natural scenes, *IEEE Pattern Analysis and Machine Intelligence, 6,* 6, 661-674.

[35] Pentland, A. (1983) Fractal-based description,*Proceedings of the International Joint Conference on Artificial Intelligence*, pp. 973-981, Karlsruhe, Germany.

[36] Medioni, G. and Yasumoto, Y. (1984) A note on using the fractal dimension for segmentation, *IEEE Computer Vision Workshop,* Annapolis, MD

[37] Pentland, A. (1984b) Shading into texture,*Proceedings of the National Conference on Artificial Intelligence*, pp. 269-273, Austin, TX

[38] Pentland, A. (1984c) Fractals: a model for both texture and shading *Optic News* October issue, p. 71

[39] Pentland, A. (1984d) Perception of three-dimensional textures, *Investigative Opthomology and Visual Science, 25,* No. 3, pp. 201.

[40] Fodor, J., (1982) *Modularity of Mind: An Essay on Faculty Psychology*, Cambridge, Mass: M.I.T. Press

[41] Gregory, R. L., (1970) *The Intelligent Eye,* New York: McGraw-Hill Book Company.

[42] Leyton, M. (1984) Perceptual organization as nested control. *Biological Cybernetics 51,* 141-153.

[43] R. Held and W. Richards (Eds.) (1975) *Recent Progress in Perception,* Readings from

Scientific American, San Francisco: W. H. Freeman and Co.

[44] Ferrie, F. P., and Levine, M. D., (1985) Piecing together the 3D shape of moving objects: an overview. In *IEEE Conference on Vision and Pattern Recognition,* San Francisco, June 19-23.

[45] Roberts, L. (1965) Machine perception of three-dimensional solids. In *Optical and Electrooptical Information Processing* Tippet, J.T., et al (Ed.) M.I.T. Press.

[46] Winston, P.H., (1975) Learning structural descriptions from examples, Ph.D. Thesis, in *The Psychology of Computer Vision,* Winston, P.H. (Ed.), New York: McGraw-Hill.

[47] Winston, P., Binford, T., Katz, B., and Lowry, M. (1983) *Proceedings of the National Conference on Artificial Intelligence (AAAI-83),* pp. 433-439, Washington, D.C., August 22-26.

[48] Davis, E. (1983) The MERCATOR representation of spatial knowledge. *Proceedings of the Eighth International Joint Conference on Artificial Intelligence,* pp. 295-301, Karlsruhe, West Germany, August 8-12.

[49] Selizski, Richard, (1985) Inverting the fractal generation process, *in preparation*

[50] Burt, P. J., and Adelson, E. H., (1983) The Laplacian pyramid as a compact image code, *IEEE Transactions on Communications, COM-31,* 4, 532-540.

[51] Reeves, W. T., (1983) Particle systems - a technique for modeling a class of fuzzy objects, *ACM Transactions on Graphics 2,* 2, 91-108.

Appendix I

The Terrain-Calc System

By: Lynn H. Quam

I THE TERRAIN-CALC SYSTEM

A. Overview

Terrain-Calc is a system for synthesizing realistic sequences of perspective views of real-world terrain that is described by a database consisting of geometric and photometric models. The geometry of the surface is described by a digital terrain model, which is a 2-dimensional array of elevations defined on a regular grid. The photometry of the terrain is described by a source image covering all or part of the area contained in the terrain model. This image is geometrically related to the terrain model by a projection (usually a perspective projection) that relates world coordinates to image coordinates.

The image-synthesis process is approximately equivalent to the following physical analogue:

(1) Create a physical model of the terrain using a construction material that has a Lambertian reflectance function.

(2) Project the source image onto the terrain model using a projector with proper focal length, placed at the proper position and orientation (equivalent to the perspective projection model relating the source image to the terrain).

(3) View the physical terrain model with a camera having the desired focal length, position, and orientation.

Views constructed according to this description are approximately what would have been seen by a camera as defined by (3) over the actual terrain at the same time that the source image was acquired. The differences are due to the following effects:

- The geometric and photometric models are limited in resolution and accuracy.
- Portions of the surface that should be visible in the synthesized view were not visible in the source image.
- The actual surface materials do not obey Lambert's Law.

The view-synthesis algorithm is related to a technique developed by the computer graphics community called "texture mapping" (Quam 1971), (Blinn 1978), (Catmull 1980). A novel algorithm is used in Terrain-Calc to avoid aliasing that results from violating the sampling theorem.

B. The Models

The geometry of the surface is described by a digital terrain model that is a 2-dimensional array of elevations z(x,y), where x and y are defined on a regular grid. Each square of the grid is cut into two planar triangular facets, choosing the diagonal that maximizes the angle between the normals to the two triangular facets.

The photometry of the terrain is defined by a digitized source image covering all or part of the geometric model. It is assumed that the surface materials obey Lambert's Law, which makes it possible to generate relatively realistic views without detailed modeling of the surface materials.

The relationship between digitized pixel values in the source image and real-world luminous flux at the surface is generally unknown because of the many parameters in the film processing chain before the image is digitized and because of the effects of atmospheric scattering. For images acquired with calibrated sensors, it would be possible to synthesize views where the light source is at a position different from that in the source image, so long as the terrain obeys Lambert's law.

C. The View-Synthesis Algorithm

Views are synthesized by iterating over the triangular facets in the terrain model, projecting the vertices of each triangle to the source and view images, and "warping" each triangular patch in the source image into its corresponding patch in the view image.

The warp step iterates over pixels on the regular grid of the synthesized view that are within each triangle, computing the position of the corresponding pixels in the source image using the linear transformation that maps the triangle in the synthesized view into the source image.

The "warp" operation starts by determining the sampling relationship between pixels in the synthesized view and pixels in the source image. For each triangle in the synthesized view, a circle of one pixel diameter is constructed at any point in the triangle. Since all of the triangles are planar, and the following projections do not include perspective scale change, the particular choice of point does not matter. A cone is constructed by projecting this circle through the projection center of the synthetic camera. This cone is intersected with the corresponding triangular facet of the terrain model, forming an ellipse. A second cone is constructed by projecting this ellipse through the projection center of the camera for the source view. The intersection of this second cone with the image plane of the source view results in an ellipse that corresponds to the circular pixel in the synthetic view (see Figure 1).

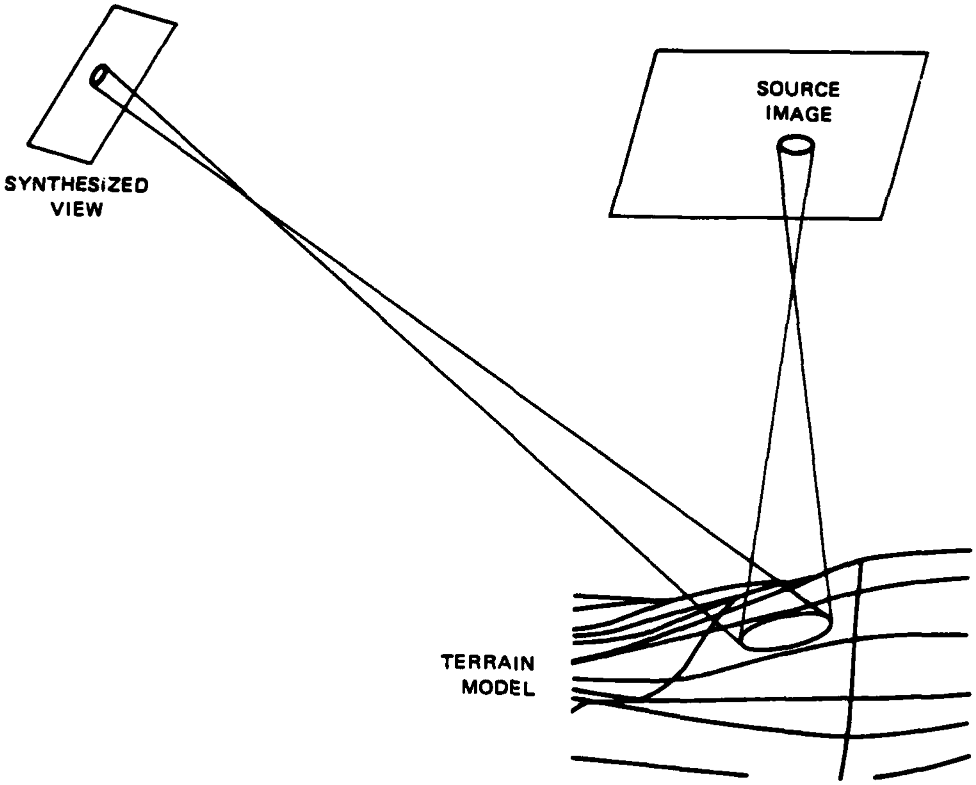


Figure 1: *GEOMETRY OF THE VIEW SYNTHESIS ALORITHM*

A somewhat more accurate, but also more complicated calculation of the sampling relationship projects a one-pixel-square area from the synthesized view, rather than a circle, and results in a quadrilateral area in the source image.

The use of this sampling relationship is essential to avoid problems due to aliasing, which result from violating the sampling theorem. To avoid aliasing, each pixel in the synthetic view is computed by integrating pixel values in the source image over an elliptical area corresponding to the pixel.

Terrain-Calc computes an approximation to the integral over an elliptical area by summing estimates of integrals over circles that have diameters approximately equal to the minor axis of the ellipse along a path corresponding to the major axis of the ellipse. The circular integrals of various diameters are formed by convolving the source image with circularly symmetric Gaussian convolution kernels of varying sizes using the hierarchical Burt algorithm (Burt, 1981).

Another form of aliasing can occur at pixels that cross occlusion edges, where pixels in the synthetic view project to several facets in the terrain model. The most severe problem of this kind occurs at occlusion boundaries, where the pixel in the synthesized view projects to widely separated facets in the terrain model. The correct calculation requires summing the intensity integrals over two or more partial ellipses corresponding to the intersections of the cone with each facet.

Hidden-surface elimination is accomplished using a variation of the "H-array" technique of Wright (Wright, 1973), which requires that (1) the facets be processed in a near-to-far order in relation to the synthetic camera, (2) that the world z-axis always project to a vertical line in the synthetic view (i.e. no roll to the camera), and (3) the geometric model be a single-valued function z(x,y). An improved algorithm (Anderson, 1982), which permits camera roll and fixes some other problems with the H-array technique, will be implemented in the near future. A more conventional z-buffer algorithm would eliminate all of the above restrictions at additional computational cost.

The time required by the view-synthesis algorithm is mainly determined by the number of pixels in the generated view and the number of facets that must be examined. For views containing approximately 320 x 250 pixels resulting from 44000 facets, the view-synthesis algorithm requires about 150 seconds on a Symbolics 3600.

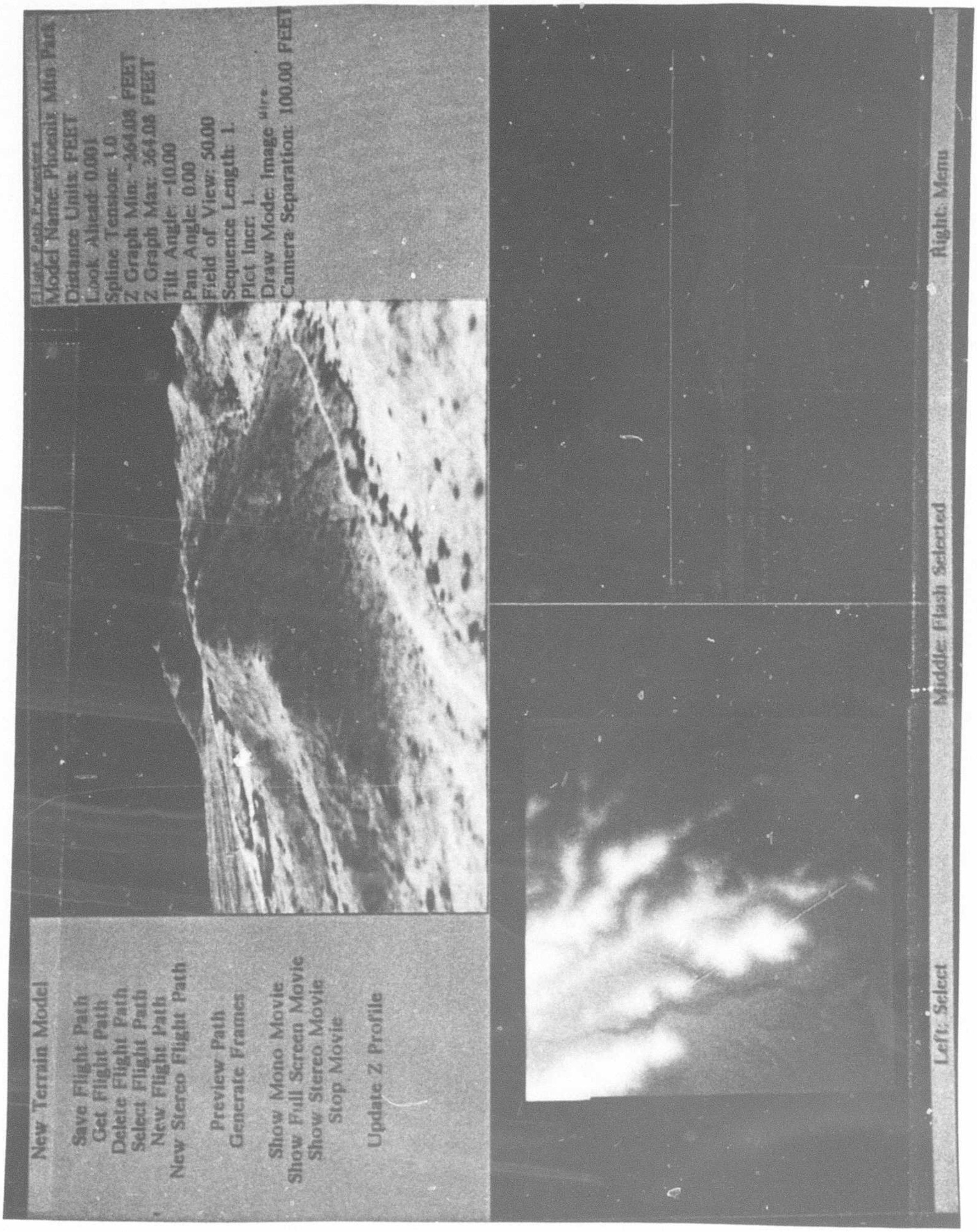
D. Interactive User Interface

Terrain-Calc also provides a sophisticated graphical interface for specifying flight paths and parameters of a simulated camera (see Figure 2).

To specify a flight path, the user first invokes an interactive curve editor to draw a curve on top of a vertical view of the terrain model as depicted on the display screen, thereby specifying the x and y components of the flight path. Terrain-Calc then displays a graph of the terrain model profile underneath the flight path, allowing the user to specify the z component of the flight path in relation to the terrain profile. A parametric spline curve is fit to the x, y, and z components of the flight path, from which position and direction can be easily computed as a function of distance along the curve.

Each synthetic view is computed by means of a perspective projection whose parameters are determined by the flight path and a parameter menu consisting of following:

- *Field of View:* Horizontal field of view. This is the angle relating the focal length to the view image width.



Flight Path Parameters

Model Name: Phoenix Mtn Park

Distance Units: FEET

Look Ahead: 0.001

Spline Tension: 1.0

Z Graph Min: -364.08 FEET

Z Graph Max: 364.08 FEET

Tilt Angle: -10.00

Pan Angle: 0.00

Field of View: 50.00

Sequence Length: 1.

Pict Incr: 1.

Draw Mode: Image wire

Camera Separation: 100.00 FEET

Figure 2: (a) TERRAIN-CALC SHOWING A SYNTHESIZED IMAGE (UPPER WINDOW), THE SOURCE DTM WITH ELEVATIONS DISPLAYED BY BRIGHTNESS (LOWER LEFT WINDOW), AND THE ELEVATION PROFILE IN THE VIEW DIRECTION (LOWER RIGHT WINDOW)

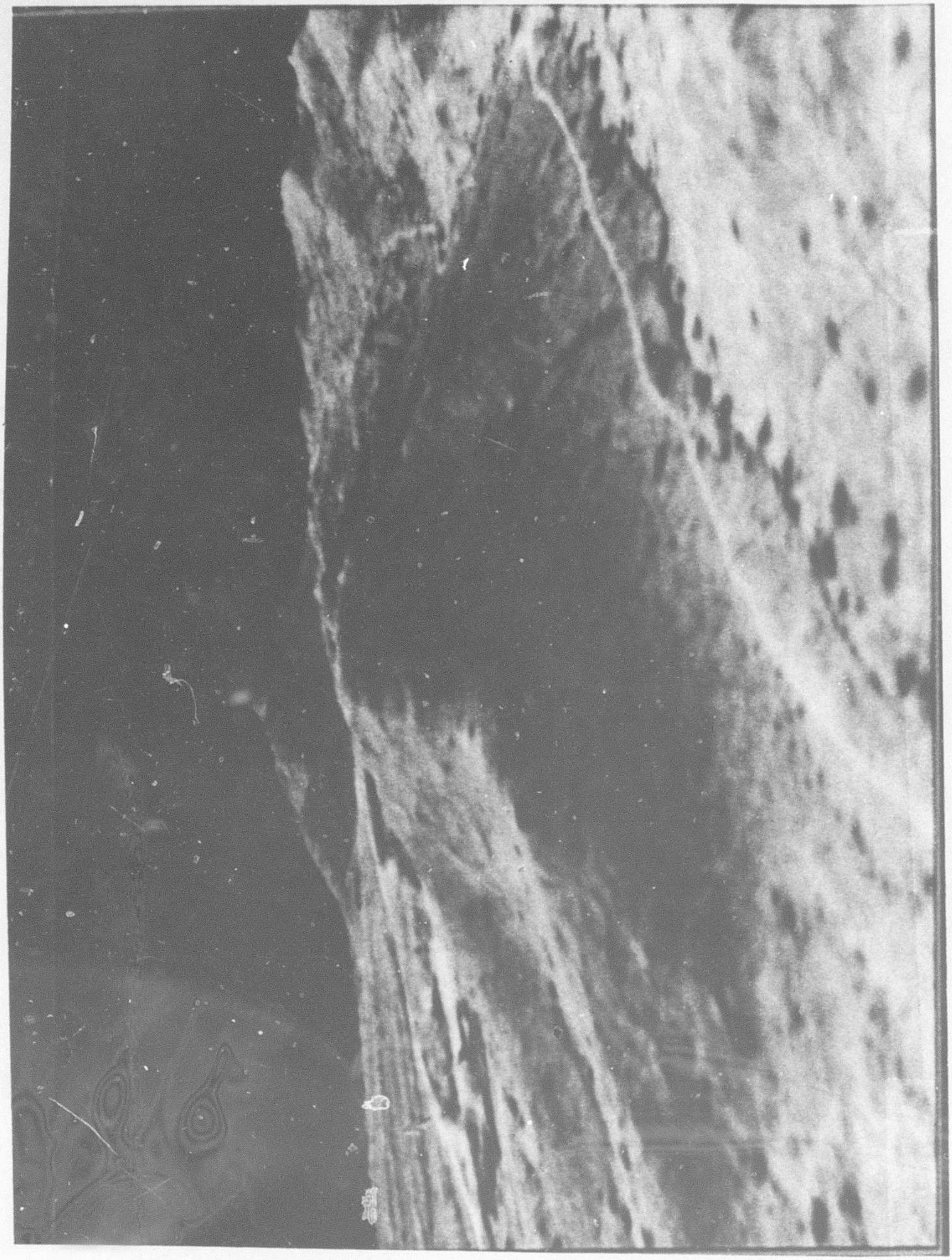


Figure 2: (b) AN IMAGE SYNTHESIZED BY TERRAIN-CALC

- *Tilt:* Tilt or pitch of the camera in a vertical plane with respect to the direction of the flight path. Currently, tilt must not be large enough to cause any rays from the camera to be exactly vertical, because of limitations of the hidden-surface algorithm.
- *Pan:* Pan or yaw of the camera with respect to the flight path.
- *Sequence Length:* Number of equally spaced views to be generated.
- *Draw Mode:* Selection of wire frame or synthetic image views.

A sequence of views spaced at equal distances along the flight path is generated. For each view, the combination of flight-path direction, tilt, and pan determines the direction of the principal camera ray, which, together with the flight-path position and focal length, determines all of the parameters of the perspective projection for the view. Sequences of views that fit in available physical memory can be dynamically displayed on the color screen at a rate of about 1.3 million pixels per second, or sixteen 320 x 250-pixel frames per second. On a Symbolics 3600 with six megabytes of physical memory, there is room for about 60 frames, each containing 320 x 250 pixels.

Stereo views are created using two identical synthetic cameras separated by a user-specified distance on a horizontal line perpendicular to the direction of the principal ray. They are displayed either as left-right pairs of images for viewing using a stereo viewing box to merge the images or as a cyan/red anaglyph image. Left-right stereo-pair sequences can be displayed at half the above frame rate, whereas anaglpyh stereo sequences can be displayed at the full frame rate.

E. Unsolved Problems and Future Directions

A major unsolved problem is how to improve the efficiency of the projection algorithm by using hierarchical terrain models, in which the level of the hierarchy (and therefore the size of the facets) is chosen in each neighborhood of the terrain model so that there are no noticeable flaws in the generated views. The use of a hierarchy is particularly important for the synthesis of oblique views, where distant facets of the terrain model project to a small fraction of a pixel in the view.

A simple hierarchical technique is to represent the terrain at a hierarchy of resolutions, obtained by convolving the terrain model z(x,y) with Gaussian kernels of various sizes and then decimating the results. The view-synthesis algorithm begins using the highest resolution in the pyramid and, for each row of facets, keeps track of the distance to the nearest facet in the row. As this distance increases, the coarser levels of resolution in the terrain hierarchy are used, in order to keep the number of view-image pixels per facet approximately constant.

This technique produces acceptable results when images are viewed in isolation, but introduces annoying artifacts in motion sequences. The problem is that the transitions between levels of the terrain hierarchy occur at different places in each image, depending on the distances between the facets and the camera. Such transitions occur in jumps. We have implemented an improvement that performs linear interpolation between levels in the resolution hierarchy to eliminate the abrupt transitions.

Currently, Terrain-Calc only handles the very restricted class of geometric models of the form $z(x,y)$. A future extension will allow a mixture of 3-D modeling techniques to be used together, using a Z-buffer (Catmull 1974) or A-buffer (Carpenter) to merge the results of the disparate modeling systems.

References

Anderson, David, "Hidden Line Elimination in Projected Grid Surfaces," ACM Transactions on Computer Graphics, October 1982, pp. 275-288.

Blinn, James, "Simulation of Wrinkled Surfaces," SIGGRAPH Proceedings, August 1978, pp. 286-292.

Burt, Peter, "Fast Filter Algorithms for Image Processing," Computer Graphics and Image Processing, May 1981, pp. 20-51.

Carpenter, Loren, "The A-Buffer, an Antialiased Hidden Surface Method," ACM Computer Graphics, July 1984, pp. 103-108.

Catmull, Edwin, "A Subdivision Algorithm for Computer Display of Curved Surfaces," University of Utah, Salt Lake City, December 1974.

Catmull, Edwin and Alvy Ray Smith, "3-D Transformations of Images in Scanline Order," ACM SIGGRAPH'80 Conference Proceedings, July 1980, pp. 279-285.

Quam, Lynn, "Computer Comparison of Pictures," Stanford Artifical Intelligence Project Memo No. 144, May 1971.

Wright, Thomas, "A Two-Space Solution to the Hidden Line Problem for Plotting Functions of Two Variables," IEEE Transactions on Computers, January 1973, pp. 28-33.

Appendix J

Parallel Guessing: A Strategy for High-Speed Computing

By: Martin A. Fischler and Oscar Firschein

PARALLEL GUESSING: A STRATEGY FOR HIGH-SPEED COMPUTATION

Technical Note No. 338

September 19, 1984

By: Martin A. Fischler, Program Director, Perception
Oscar Firschein, Staff Scientist

Artificial Intelligence Center
Computer Science and Technology Division

Presented to the Workshop on Algorithm-Guided Parallel Architectures for Automatic Target Recognition [Sponsored by the DARPA Tactical Technology Office (TTO), the Naval Research Laboratory, and the Army Night Vision and Electro-Optics Laboratory (NVL)] July 16-18, 1984 in Leesburg, VA.

SRI Project 5355
Contract MDA903-83-C-0027

CONTENTS

ABSTRACT

Attempts have been made to speed up image-understanding computation involving conventional serial algorithms by decomposing these algorithms into portions that can be computed in parallel. Because many classes of algorithms do not readily decompose, one seeks some other basis for parallelism (i.e., for using additional hardware to obtain higher processing speed). In this paper we argue that "parallel guessing" for image analysis is a useful approach, and that several recent IU algorithms are based on this concept. Problems suitable for this approach have the characteristic that either "distance" from a true solution, or the correctness of a guess, can be readily checked. We review image-analysis algorithms having a parallel guessing or randomness flavor.

We envision a parallel set of computers, each of which carries out a computation on a data set using some random or guessing process, and communicates the "goodness" of its result to its co-workers through a "blackboard" mechanism.

I INTRODUCTION

Sophisticated image analysis often requires the use of a sequence of time-consuming algorithms, such as feature extraction, region growing, and model instantiation. Such processing sequences currently require several minutes for their computations on commonly available machines, see Table 1. "Real-time" scene analysis with frame rates of 1/30 second will require three or four orders of magnitude speedup. If future computer technology advances provide us with one or two orders of magnitude over the next 5 to 10 years, two or three orders of magnitude improvement are still required for practical applications in the indicated 5-10 year period.

Table 1
Timing for Some Image Understanding Algorithms

(all timing is CPU time of a VAX 11/780)

RELAX relaxation algorithm, University of Maryland, (3x3 window, 128x128 image)	3 minutes/iteration
Phoenix segmentation algorithm, Carnegie Mellon university, 500x500 image	33 minutes
GHOUGH, generalized Hough Transform, University of Rochester (variable, depending on image size, number of rotations and radii tried, and template size)	1-5 minutes

In recent years, parallel architectures for image processing have been developed; a recent survey of these is given in [Reeves 1984] and in [Duff 1983]. These architectures are largely tailored for the natural parallelism found in convolution, filtering, and other "low-level" scene analysis processes. However, higher-level processes do not exhibit such parallelism and, in general, algorithmic parallelism cannot be achieved by attempting to decompose essentially sequential algorithms. (Shannon showed this for the case of n-dimensional switching functions, [Shannon 1949]).

We therefore seek a generally applicable formalism for image analysis algorithms that offers a natural parallelism, so that we can trade additional hardware for decreased computation time. One class of such algorithms takes advantage of the following observation: *It is often much faster to verify the* *correctness of a guess, than to compute the solution.* Based on this observation, we postulate an architecture based on a large set of processors that guess an answer (1) by means of random selection, (2) by an exhaustive "rough grain" selection, or (3) by ***intelligent*** guessing. Such guessing mechanisms become especially important in problems in which the data are noisy, or when there is not an adequate analytical model.

II APPROACH TO PARALLELISM

Our approach is, therefore, to develop image analysis algorithms suitable for parallel computation that are based on guessing a good answer. The basic idea is that each module simultaneously takes a different guess and computes a "goodness" value for the guess. When a "good" guess is made, its result and the goodness value are entered on a "blackboard." The blackboard controller indicates the basis for further iterations by constraining the range of new values to be chosen and determines when a suitable answer has been found.

In contrast to most current concepts, based on a small "grain size" and high-bandwidth communication between processing modules, we seek algorithms that do not require lockstep operation of processors and require a minimum of communication between processors. The proposed parallel architecture is shown in Figure 1. Symbolic structures derived from low-level processing are stored in a *blackboard.* Parallel processors derive their input data from this blackboard directed by a control processor. Intermediate results are returned by the processors to the blackboard. Results of the computation are analyzed by the control processor and then used as output. The requirements for algorithms to be used in this architecture are as follows:

- **Selection**. A selection method for the guess must be provided, using intelligent guessing, a random selection, or exhaustive selection from a roughly quantized space. The selection process can be carried out in *data space* by selecting from among the input data, in ***parameter*** space in which there is a selection of values for one or more of the model parameters, or in both data and parameter space.
- **Goodness of result**. There must be a simple measure of the goodness of the result obtained using the guess.
- **Control**. Some method must be provided for selecting the best current guesses, for using the best current guess to constrain additional guessing, for accomplishing efficient guessing by partitioning the space of guesses, and for determining when the overall process is to stop.

III EXAMPLE ALGORITHMS

Many existing scene-analysis-related algorithms can be viewed as satisfying the above requirements. The Hough transform, RANSAC, back-projection techniques, branch-and-bound, and functional optimization are particular examples. These are described below.

A. Hough Transform Approach

The Hough transform [Duda and Hart, 1972] can be used when little is known about the scale and location of the boundary of an object we wish to find, but its shape can be described by a parametric curve, e.g., a straight line. This class of algorithms finds the parameters of a model within a roughly quantized range of variable values in the equations of the model. For example, if we are given a list of edge pixels and wish to find an acceptable straight line that passes through or near many of these pixels, we can use the approach shown in Figure 2.

This approach is mechanized using random data selection. Each processor selects an edge point from the list and considers a span of lines of various directions through the point. The distance of the normal to each such line is computed, and the value of the normal distance and angle for each line is used to increment an appropriate *(angle, normal distance)* histogram "bucket" on the blackboard. A control processor associated with the blackboard stops the process when it determines that a histogram peak is evident and returns the *(angle, normal distance)* value of the peak as the parameters of the desired line.

1. *RANSAC*

The random selection and consensus (RANSAC) approach [Fischler and Bolles, 1981] is a procedure that uses a random selection of exactly enough data points to satisfy a model. Each trial involves random point selection and testing of the proposed model on the remaining points. A simple example of RANSAC is the case of determining an acceptable line, given a set of candidate edge points. A pair of points is randomly selected, as shown in Figure 3, and the sum of the absolute values of the deviations of the other points from this line is used as a measure of goodness of fit.

The algorithm uses random selection of data points. In the parallel mechanization, each processor selects a pair of input points, computes the line parameters, and then determines the sum of the deviations of the other points from this line. Each processor looks at the blackboard to see if the sum of the deviations obtained is less than the best value posted on the blackboard. If it is, then the previous value is replaced by the new value and the line parameters. When the sum of the deviations is less than a desired amount (and the individual deviations are not correlated in any way), the control processor stops the process by writing a termination message on the blackboard.

2. *Back Projection*

In the "back-projection" problem, we are given an image and want to determine the structure of the scene that produced the image. Witken [Witken 1981] finds the 3-D orientation of small planar patches within the scene to obtain an estimate of the 3-D geometry of objects in the scene. His approach makes the following assumption: ***An arbitrary scene will have no favored direction for its visible edges.*** The algorithm first finds the edges in the image and then finds the tangent lines to these edges. A trial and error procedure of assuming specific planar patch orientations in the scene is now carried out. The "best" planar patch for each local region in the scene is the one for which the distribution of the "back-projected" line orientations will be "most random."

In the parallel mechanization shown in Figure 4, each processor obtains, from the blackboard, a list of tangent lines in some local patch of the image. Each processor accepts a complete set of scene planar patch orientation parameters specified on the blackboard and uses these parameters to back-project the tangent lines Each processor develops a histogram of line directions, for each trial orientation of the scene patch, as an indicator of the randomness for the line directions: the flatter the histogram, the better the estimate. Each processor reports to the blackboard the best orientation of the patch it is analyzing and the goodness of the result. The blackboard control computer specifies the range of plane orientations to use, assigns portions of the image to the computers for analysis, and decides when to stop the process.

3. *Branch-and-Bound*

Branch-and-bound is a popular technique that has been used successfully in the solution of problems that arise in combinatorial optimization and artificial intelligence. In a branch-and-bound approach, the solution space is organized as a graph that is usually a tree, with each link representing a value. The goal is to proceed from the root node to some end node in a way that maximizes or minimizes the sum of the path values. Various forms of branch-and-bound are all based on the idea of avoiding paths that are unproductive. Thus, one might begin by following a random path from the root to the goal to obtain a total cost C for that path. One then explores another path, stopping the exploration of that path when the path cost exceeds C. If a lower-total-cost path from root to goal is found, then its cost becomes the new C. Parallelism can be introduced into the process by expanding more than one path during each iteration. A parallel computer for implementing branch-and-bound algorithms is given in [Wah and Ma 1982], and the problems that arise in parallel branch-and-bound are given in [Lai and Sahni 1984].

Using the multiple-path-expansion approach, each processor follows a path, computing the cumulative cost as it proceeds. When a processor has followed a path from root to goal, it posts the total cost on the blackboard. Any processor that currently has a greater cost terminates its current path and pursues a new path. This procedure is shown in Figure 5.

4. *Maximum or Minimum of a Function*

There are many image-analysis applications involving the determination of the maximum or minimum of a function. If the function is relatively smooth, then an iterative gradient approach can be used in which a measure of the gradient is used to determine the next guess as to the independent variable. If the function has many local maxima or discontinuities, a *coarse-fine* approach is more appropriate, in which random exploration is carried out in coarse partitions of the independent variable, and a finer exploration is then made in locations that seem promising.

In the *coarse-fine* parallel mechanization, each processor is assigned a range and makes random guesses of the independent variable within its assigned range. After n random looks, only the most promising ranges are retained, and the processors are redistributed to cover the selected ranges. The random guessing procedure is continued for a number of iterations in the selected ranges, and subranges are identified for further exploration. The motivation is to avoid getting trapped in a local maximum at an early stage of the process. In the parallel mechanization shown in Figure 6, we use a simple one-dimensional example of finding a global maximum, given a "noisy" function having many local maxima.

IV CONCLUSIONS

Guessing techniques based on randomness or exhaustive bucketing can be important in image analysis, since such guessing is often needed in the face of data errors or lack of a suitable analytic model. In addition, guessing offers the advantage of a uniform approach to achieving parallelism. We have indicated a parallel architecture that can take advantage of such an approach and some present-day algorithms that can be viewed from this point of view. Rethinking of some of the "classical" image-analysis algorithms in this context can prove fruitful.

REFERENCES

Duda, R.O. and P.E. Hart, "Use of the Hough transformation to detect lines and curves in pictures," Comm ACM, 15,1, Jan. 1972, 11-15.

Fischler, M.A. and R.C. Bolles, "Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography," Comm. ACM Vol. 24(6), pp. 381-395 (June 1981).

Frisby, J.P., "Seeing: Illusion, Brain, and Mind," Oxford University Press, 1980.

Lai, T. and S. Sahni, "Anomalies in Parallel Branch-and-Bound Algorithms," Comm. of the ACM, June 1984, Vol. 27, No. 6.

Reeves, A. P., "Parallel Computer Architectures for Image Processing," Computer Vision, Graphics, and Image Processing 25, 68-88(1984).

Shannon, C.E., "Synthesis of two-terminal switching networks," Bell System Technical Journal, Jan. 1949 28(1):59-98.

Wah, B. and Ma, Y., "NANIP - A parallel computer system for implementing branch-and-bound algorithms." Proc. 8th Ann. Symp. on Computer Architecture, 1982, pp. 239-262.

Witkin, A. P., "Recovering surface shape and orientation from texture," Artificial Intelligence 17, 1981, 17-47.

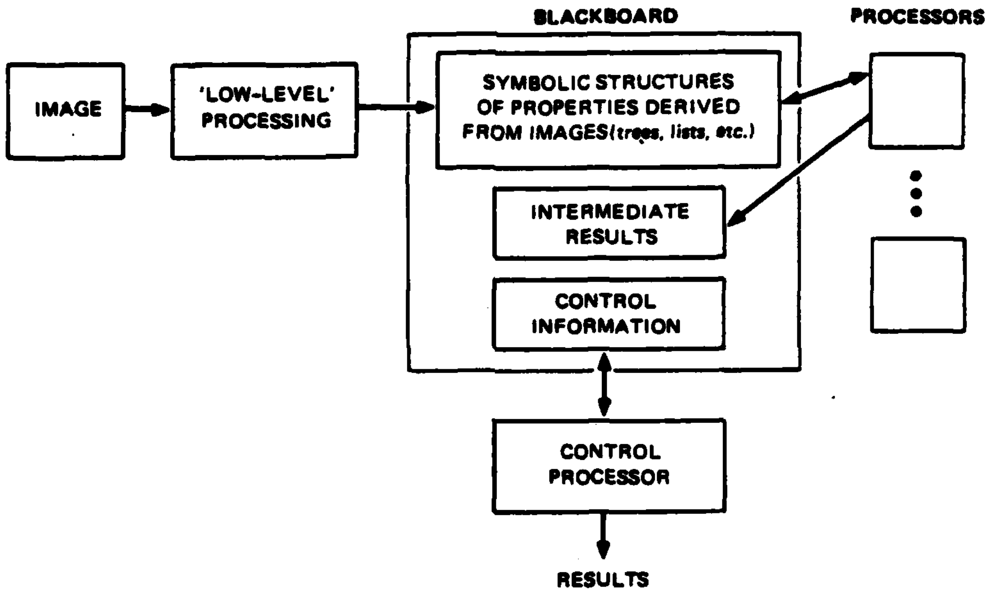
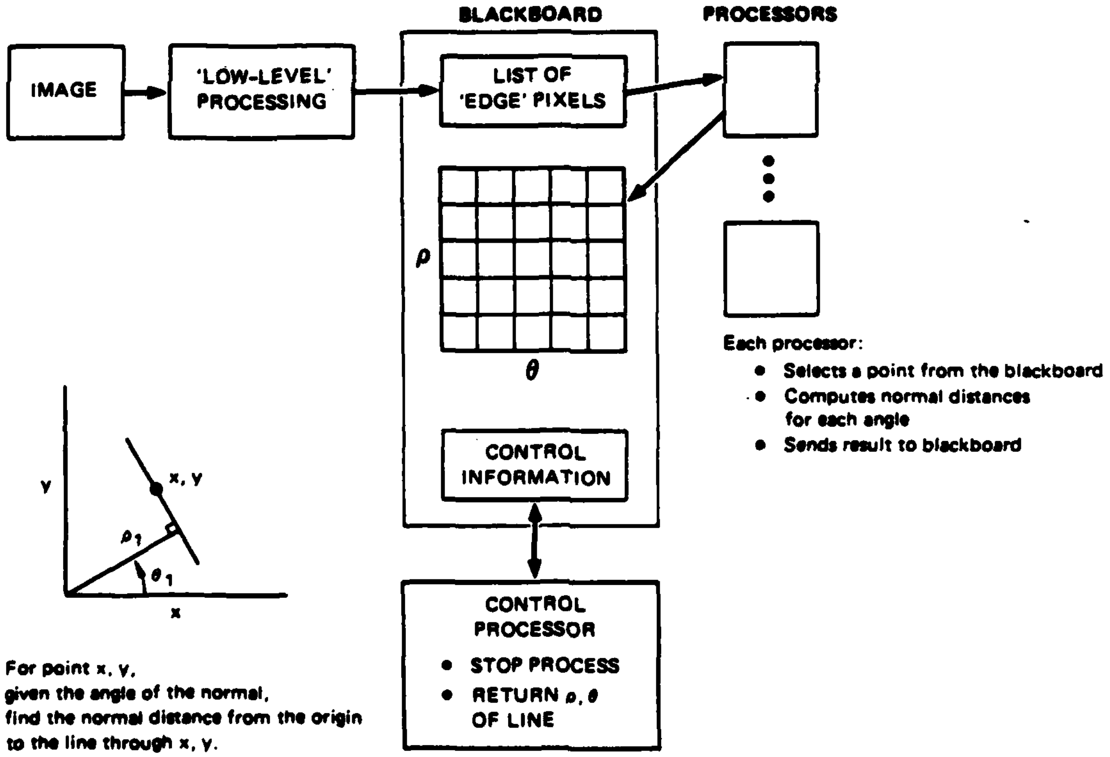


FIGURE 1 PARALLEL PROCESSING ARCHITECTURE FOR GUESSING ALGORITHMS



Each processor:

- Selects a point from the blackboard
- Computes normal distances for each angle
- Sends result to blackboard

For point x, y,
given the angle of the normal,
find the normal distance from the origin
to the line through x, y.

FIGURE 2 HOUGH APPROACH

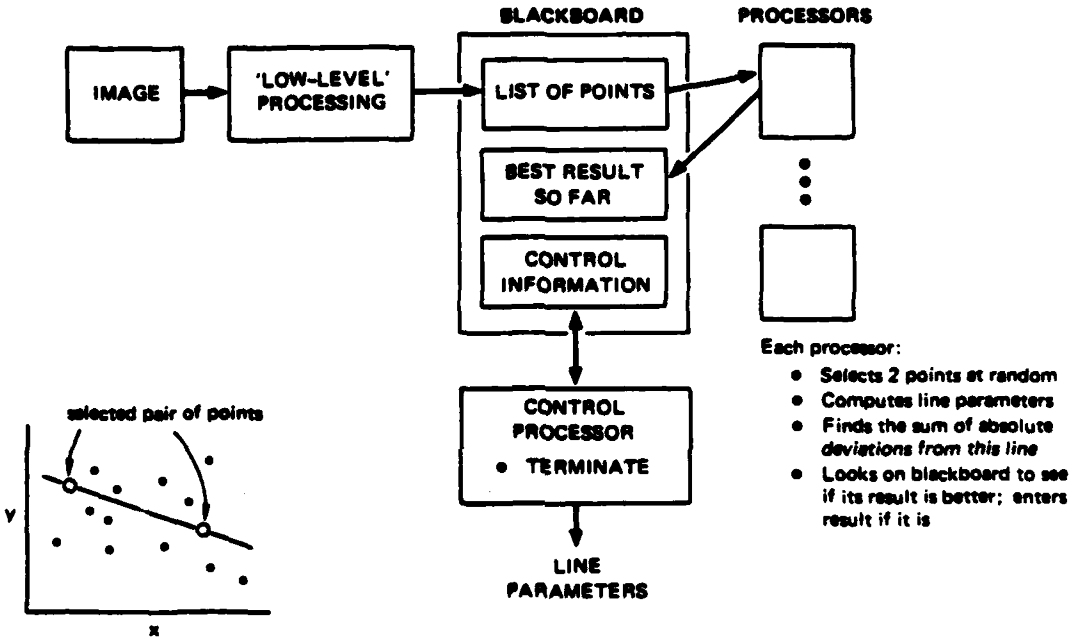

BLACKBOARD
PROCESSORS
IMAGE
'LOW-LEVEL' PROCESSING
LIST OF POINTS
BEST RESULT SO FAR
CONTROL INFORMATION
CONTROL PROCESSOR
• TERMINATE
LINE PARAMETERS
Each processor:
• Selects 2 points at random
• Computes line parameters
• Finds the sum of absolute *deviations from this line*
• Looks on blackboard to see if its result is better; enters result if it is
selected pair of points
Y
x

FIGURE 3 RANSAC:RANDOM SELECTION AND CONSENSUS

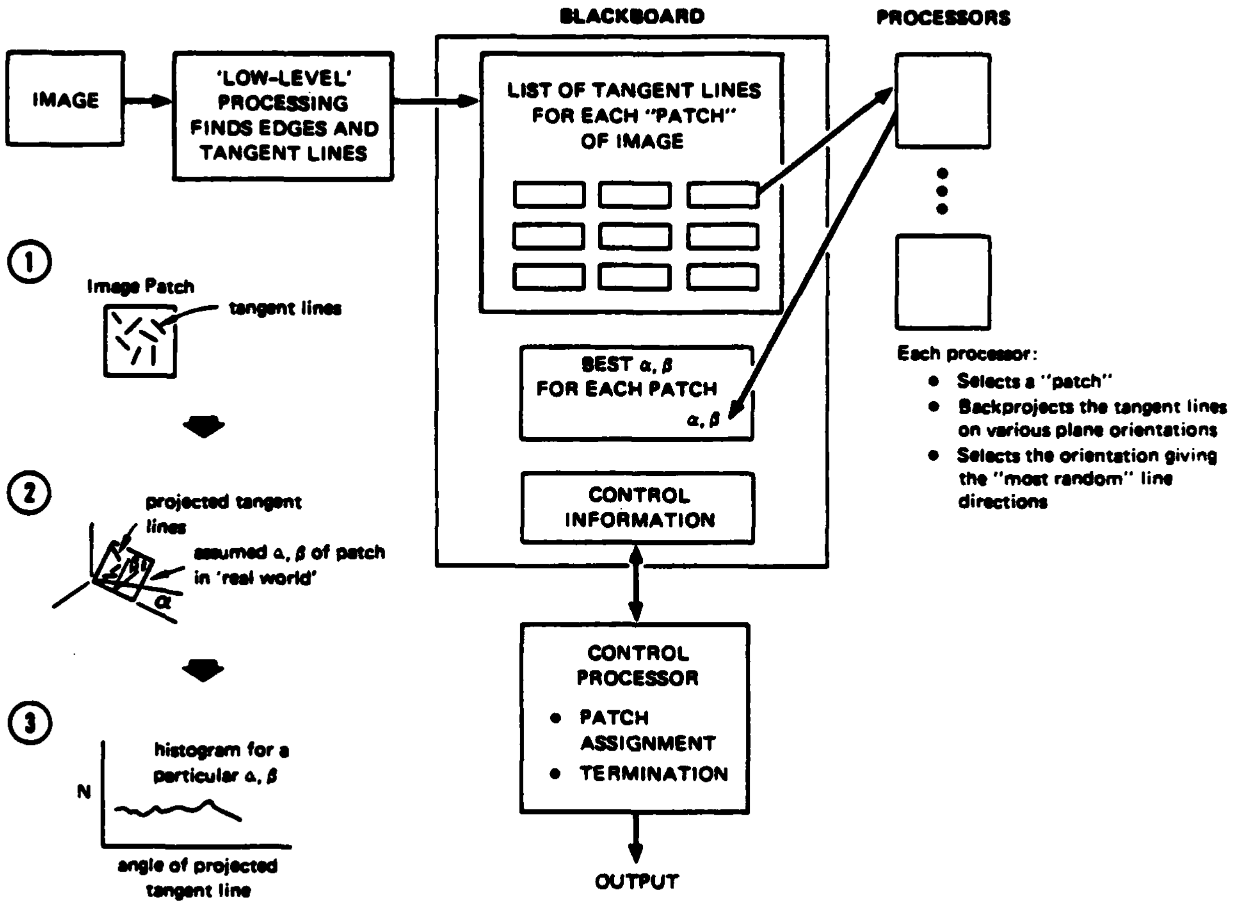

BLACKBOARD
PROCESSORS
IMAGE
'LOW-LEVEL' PROCESSING FINDS EDGES AND TANGENT LINES
LIST OF TANGENT LINES FOR EACH "PATCH" OF IMAGE
BEST α, β FOR EACH PATCH
α, β
CONTROL INFORMATION
Each processor:
• Selects a "patch"
• Backprojects the tangent lines on various plane orientations
• Selects the orientation giving the "most random" line directions
1
Image Patch
tangent lines
2
projected tangent lines
assumed α, β of patch in 'real world'
α
3
histogram for a particular α, β
N
angle of projected tangent line
CONTROL PROCESSOR
• PATCH ASSIGNMENT
• TERMINATION
OUTPUT

FIGURE 4 BACK-PROJECTION

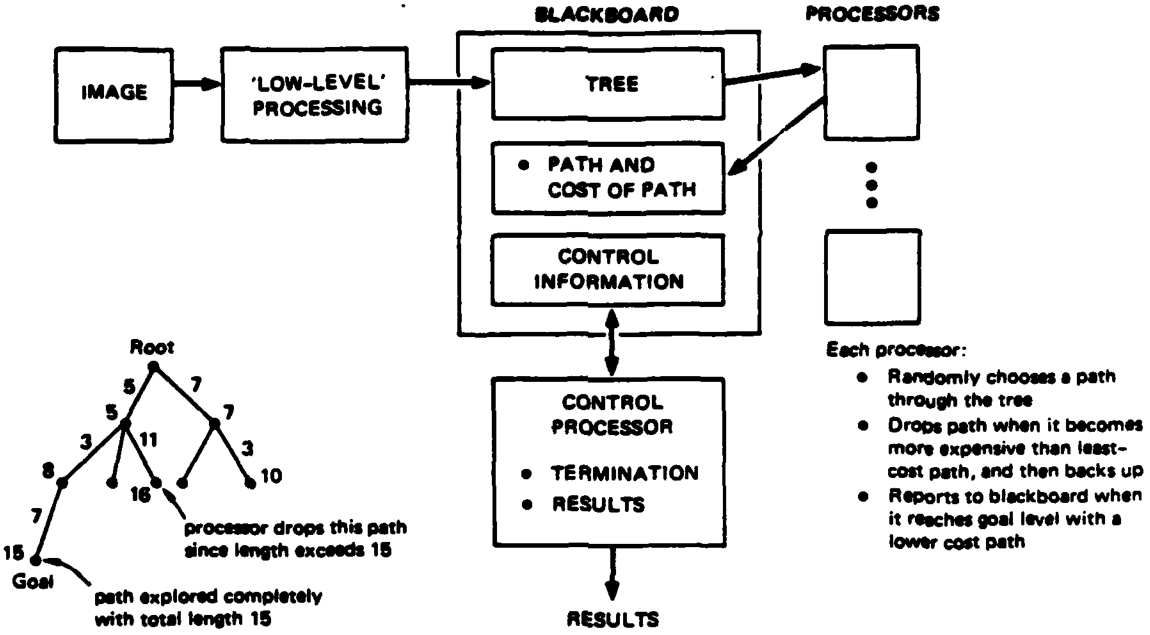

BLACKBOARD
PROCESSORS
IMAGE
'LOW-LEVEL' PROCESSING
TREE
• PATH AND COST OF PATH
CONTROL INFORMATION
CONTROL PROCESSOR
• TERMINATION
• RESULTS
RESULTS
Root
5
7
5
7
3
11
3
8
10
16
7
15
Goal
processor drops this path since length exceeds 15
path explored completely with total length 15
Each processor:
• Randomly chooses a path through the tree
• Drops path when it becomes more expensive than least-cost path, and then backs up
• Reports to blackboard when it reaches goal level with a lower cost path

FIGURE 5 BRANCH AND BOUND

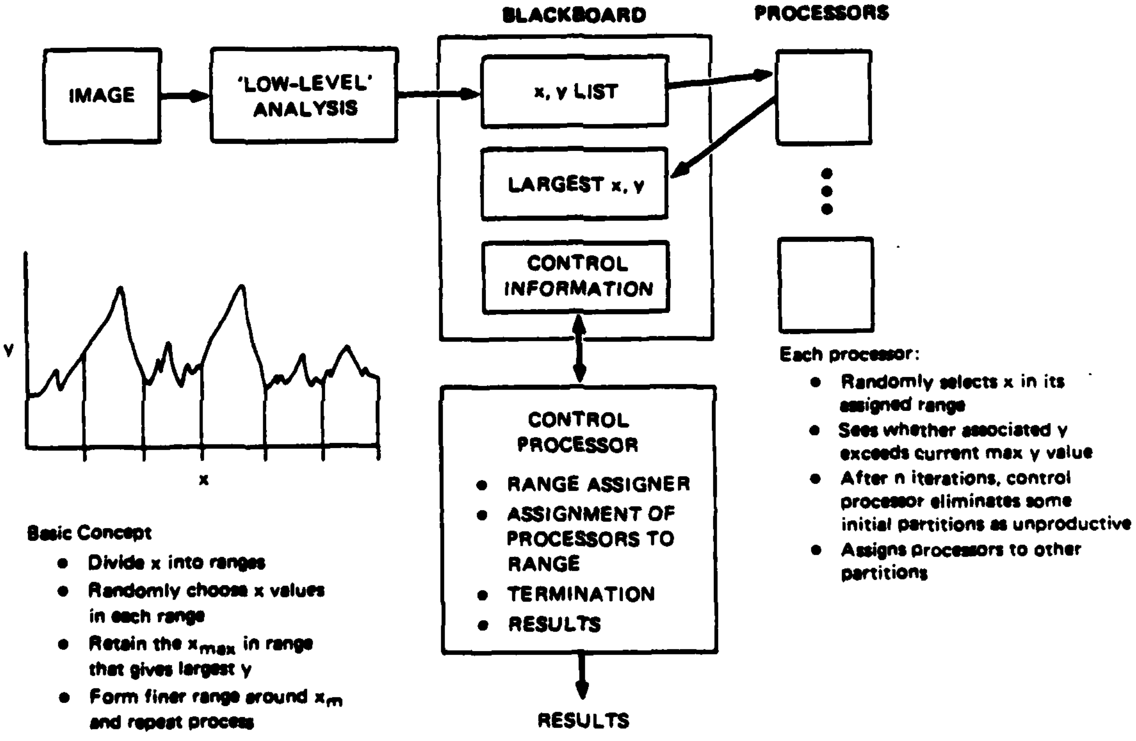

BLACKBOARD
PROCESSORS
IMAGE
'LOW-LEVEL' ANALYSIS
x, y LIST
LARGEST x, y
CONTROL INFORMATION
y
x
CONTROL PROCESSOR
• RANGE ASSIGNER
• ASSIGNMENT OF PROCESSORS TO RANGE
• TERMINATION
• RESULTS
RESULTS
Basic Concept
• Divide x into ranges
• Randomly choose x values in each range
• Retain the x_max in range that gives largest y
• Form finer range around x_m and repeat process
Each processor:
• Randomly selects x in its assigned range
• Sees whether associated y exceeds current max y value
• After n iterations, control processor eliminates some initial partitions as unproductive
• Assigns processors to other partitions

FIGURE 6 COARSE-FINE, MAX-MIN OF A FUNCTION

ADDITIONAL DISTRIBUTION

Defense Documentation Center 12 copies
Cameron Station
Alexandria, VA 22314

Defense Logistic Agency letter only
DCASMA, San Francisco
1250 Bayhill Drive
San Bruno, CA 94066

Director 1 copy
Defense Advanced Research
Project Agency
ATTN: Program Management
Lt.Col. Robert Simpson, Jr.
1400 Wilson Blvd.
Arlington, VA 22209-2308

D. Rusco 1 copy
STT
DMA Aerospace Center
3200 South 2nd Street
St. Louis, MO 63118

W. Stahler 1 copy
DMAHTC
6500 Brookes Lane
Washington, DC 20315

Jack Wallace 1 copy
DMAAC/STT
3200 South 2nd Street
St. Louis, MO 63118

Dr. Charles F. Martin, Chief 1 copy
Advanced Technology Division
Defense Mapping Agency
Bldg. 56, U.S. Naval Observatory
Washington, DC 20305

Dr. Bruce Waxman 1 copy
Defense Mapping Agency
Bldg. 56, U.S. Naval Observatory
Washington, DC 20305

Dr. Robert Leighty 1 copy
USAETL
Telegraph & Leaf Roads
Fort Belvoir, VA 22060

Lt.Col. Jay Larson
Defense Mapping Agency
Bldg. 56, U.S. Naval Observatory
Washington, DC 20305

Bill Alford
Defense Mapping Agency
Bldg. 56, U.S. Naval Observatory
Washington, DC 20305